IN-37

NASA GRANT REPORT

NASA GRANT NAG8 - 242

CONTROLLER REDESIGN FOR THE HUBBLE SPACE TELESCOPE

Prepared by:

R. Dennis Irwin
Russell D. Glenn
W. Garth Frazier
Douglas A. Lawrence
Randolph F. Follett

On behalf of:

Department of Electrical and Computer Engineering
College of Engineering and Technology
Ohio University
Athens, Ohio 45701

September 1993

Final Report

Prepared for:

Pointing Control Systems Branch
NASA George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TABLE OF CONTENTS

1.0 INTRODUCTION

The Pointing Control System (PCS) onboard the Hubble Space Telescope (HST) is required to maintain line of sight pointing error less than 0.007 arc-seconds for observations lasting as long as 24 hours. Since deployment, an unanticipated disturbance source has resulted in line of sight jitter far exceeding this requirement. Detailed analysis has indicated that this disturbance is most likely due to thermally induced in-plane and out-of-plane flexing of the solar arrays (SA) at frequencies of roughly 0.6 Hz and 0.11 Hz, respectively. NASA efforts to redesign the HST PCS have led to significant, yet insufficient, pointing error attenuation [1].

The purpose of this report is to detail efforts in applying Multi-Input, Multi-Output (MIMO) analysis and design techniques to the PCS redesign problem. In particular, the notion of singular value frequency response (i.e., H^∞ specifications) is used extensively in formulating performance specifications and performing stability analyses for coupled MIMO systems such as the HST.

The remainder of this report is organized as follows. Section 2 contains the control problem definition,. simulated results of PID and SAGA-II controller performance, and a discussion of the goals of the controller redesign effort. Also contained in Section 2 is a short discussion of the design approach taken to achieve these goals.

Section 3 addresses the modeling issues involved in this effort. Specifically, the alternative models originally available for design and simulation are presented, evaluated, and compared. It is argued that these models, particularly the composite modal models, are insufficient for reliable control system design or controller evaluation and that the only available MIMO models are insufficient for verification of control system designs via simulation. Another model built from modal gain product matrices proved to be more reliable for simulation and more robust for control system design. This improved model is also presented.

Section 4 presents a derivation of the method used for simulating controller performance. Also included are discussions of how simulations were performed using recorded flight data. These data were used to compare the redesigned controllers to the proportional-integral-derivative (PID) controller in place at the time the data was recorded. Purely linear simulations provided the initial evaluation of controller performance; subsequent simulations involving vehicle torque limits and fixed point arithmetic implementation provide a more stringent verification of the controllers. Since the original fixed point implementation was customized for the SAGA-II controller, details of our new and more general fixed point arithmetic implementation are included.

Section 5 contains a summary of attempts to design controllers using analytical H^∞ design techniques. Included are discussions of problems encountered with various HST plant models and the efforts undertaken to acquire a suitable design model, the loop shaping philosophy applied in the design, the steps necessary to alleviate the numerical difficulties encountered, and the results of a successful design.

Section 6 is a detailed description of the hybrid numerical/H^∞ approach that was used for designing a reduced-order controller. Numerical methods are utilized to achieve MIMO closed loop H^∞ specifications. First, the design philosophy is discussed, followed by a discussion of the existing SAGA-II flight controller, including an analysis of stability margins and simulated performance. Finally, the numerically obtained design is discussed and stability margins and simulated performance are compared to those of the SAGA-II controller.

Section 7 presents conclusions and recommendations.

2.0 HST CONTROL REDESIGN PROBLEM DEFINITION

A pictorial representation of the HST is given by Fig. 2.1. In this report the axis labeled V_1 is referred to as the roll axis, V_2 is the pitch axis, and V_3 is the yaw axis. A simplified block diagram of the PCS is given in Fig. 2.2. It is important to note that Fig. 2.2 contains only the "gyro hold" portion of the PCS. There are actually 1 Hz outer attitude loops that utilize the Fine Guidance Sensor. However, redesigns previously carried out by NASA/Lockheed did not address the issue of redesigning these loops; hence for comparison purposes only the gyro hold portion is considered here, as well. From Fig. 2.2 it is clear that the gyro hold loops are a three-input, three-output 40 Hz sampled-data system with an 8 millisecond computational delay. The blocks labeled $T_{W/V}$ and $T_{V/G}$ are, respectively, the body coordinate to reaction wheel assembly (RWA) transformation matrix and the rate gyro assembly (RGA) to body coordinate transformation matrix. The block labeled G_p represents the HST plant transfer function matrix, G_A is the controller, I_V is the HST inertia matrix, and PID is the original PID controller. Effects of the solar array disturbances with the original PID controller are seen in the flight data derived power spectral density (PSD) of Fig. 2.3, where it is evident that the predominant disturbances are at roughly 0.1 Hz and 0.6 Hz. These are thought to be primarily due to thermally induced out-of-plane and in-plane solar array bending modes, respectively.

The possibility that the disturbance sources are not reliably modeled also impacts on the problem of creating a design simulation. This problem has been avoided by NASA via the use of a trick in which the flight data is "played back" through a certain modified linear system for which the response is precisely that which would occur with the redesigned controller implemented with the unknown disturbances. This is valid only when the HST plant model is perfectly known and the unknown disturbances enter the plant as pure torque disturbances. Essentially, the method "backs out" the disturbances which must have caused the flight data response with the PID controller and then uses this identified disturbance "model" to simulate the response of the HST when other controllers are implemented. The details of this simulation approach are given in Section 4.

Using this approach to simulation, the predicted response of the SAGA-II controller is given in the form of a representative (V_3) PSD by Fig. 2.4. Clearly, in comparison to Fig. 2.3 the SAGA-II controller has achieved significant attenuation of the 0.1 Hz and 0.6 Hz disturbances. This result is verified by the preliminary flight data reported in Ref. 1.

For the present redesign effort the following are goals:

1. Maintain the per-axis LOS jitter below .005 arcseconds throughout a 500 second simulation that includes a day to night terminator crossing.

2. Maintain system type at three in each axis in order to provide rejection of low

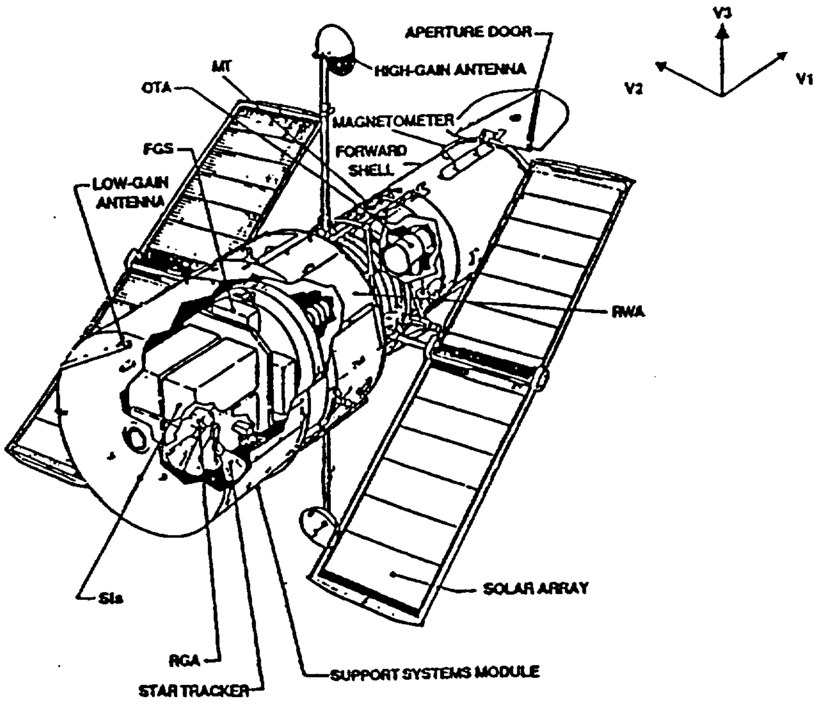

Figure 2.1 Hubble Space Telescope

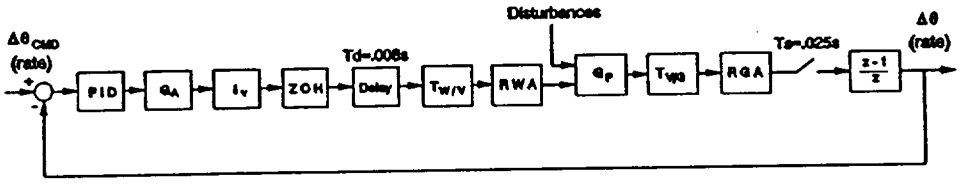

Figure 2.2 Simplified PCS Gyro Hold Block Diagram

frequency gravity gradient and aerodynamic disturbance torques and also to maintain the gyro hold controller interface to the outer attitude loops.

3. Maintain the jitter specification of requirement 1 in the face of 0.8 N-m torque limits in RWA coordinates.

4. Maintain the jitter specification when implemented in 24 bit fixed point arithmetic using 13 bits to the right of the radix point.

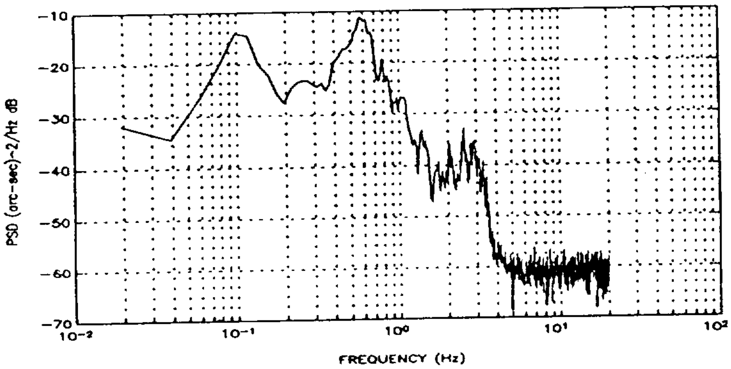

Figure 2.3 Representative Attitude PSD for the Original HST PID Controller (Flight Data)

The design approach reported here is primarily that of attempting to reduce the attitude PSD over the frequency range 0 Hz to roughly 3 Hz, i.e., over those frequencies at which the SAGA-II controller PSD exhibits significant signal power density. Simultaneously, the MIMO stability margins will be enhanced. Performance comparisons are also made on the basis of simulated RMS pointing error statistics and peak pointing error statistics.

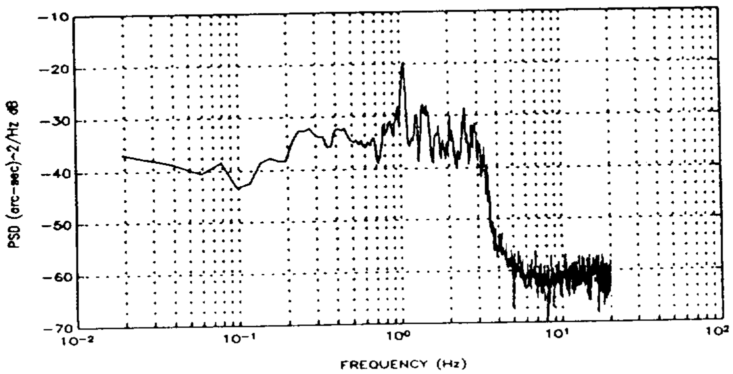

Figure 2.4 Representative (V_3) SAGA-II Simulated PSD

3.0 MODELING ISSUES

Several sets of HST models were provided by NASA in support of the PCS redesign effort. Ref. 1 contains pre-launch and post-launch composite SISO modal models given in terms of the three body coordinates. Also provided were models derived from the TREETOPS computer program. Finally, after receiving requests for a MIMO coupled modal model, NASA sent documentation[2] which contains sets of modal gain product matrices from which models for three different solar array angles may be constructed. This section discusses each type of model in detail.

3.1 Composite SISO Modal Models

Reference 1 contains two sets of modal gain coefficients for composite SISO modal models; a set of pre-launch gains and a set of post-launch gains. The models are composite in the sense that they contain modes for several solar array orientations within the same model. Since the models contain modes for various solar array angles, they do not accurately represent all of the properties of the plant under any particular solar array orientation. As such, they are not well suited for MIMO design purposes, since the stability properties of any controllers designed from such models are suspect. Furthermore, the gain coefficients are presented for three independent SISO models, one for each of the three vehicle body coordinate axes. Insufficient information is provided to determine coupling between the axes, making construction of a coupled MIMO model impossible. Since the purpose of the design effort is to apply MIMO controller design techniques to the HST, a coupled MIMO model is essential. In light of these facts, no attempts at controller design have been undertaken with these SISO models. Fig 3.1 shows a sample magnitude response from the post-launch composite SISO modal model.

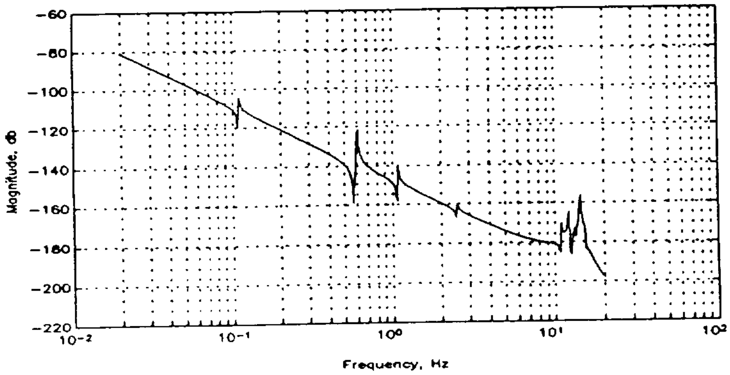

Figure 3.1 Representative (V_3) Post-Launch Composite Modal Model Magnitude Response

3.2 TREETOPS Models

Four continuous-time HST models derived from the TREETOPS computer program were provided by NASA in the original HST PCS redesign information package. The four models represent the 0°, 45°, 90° and 105° solar array orientations. The following figure displays a representative magnitude response plot from the 90° TREETOPS model.

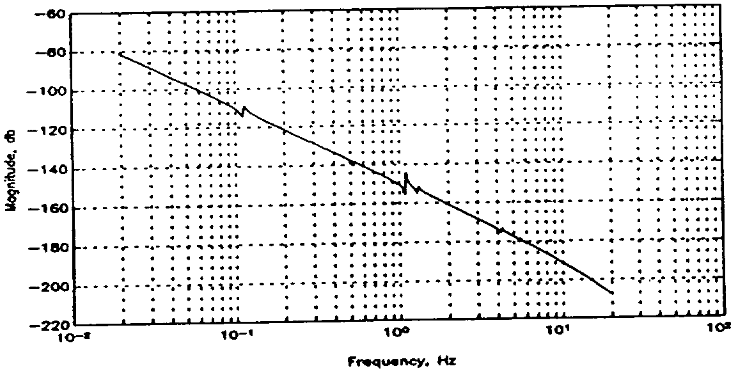

Figure 3.2 Representative (V_3) TREETOPS Model Magnitude Response

The models are 118th order; the 0° and 105° models have 28 inputs and 6 outputs, and the 45° and 90° models have 24 inputs and 6 outputs. The inputs are the control torques in the three vehicle axes, and various disturbance torques fabricated when the models were generated with the TREETOPS software. The six outputs are the angular rates and positions in the 3 vehicle axes. A MATLAB® M-file provided by NASA separates the large TREETOPS realization into the system A matrix, a control input distribution matrix, a disturbance input distribution matrix, a state to angular position output distribution matrix, and a state to angular rate output distribution matrix. The realization is then discretized and a .008 second computational delay is modeled by a first degree Padé approximation. The program ran successfully as provided by NASA with the 0° and 105° models, but modifications were required to allow it to function with the 45° and 90° models, due to the differing number of inputs. Queries to NASA regarding the reason for the differing number of disturbance inputs in the various models remain unanswered.

The philosophy applied to controller design is to achieve high controller gain in the disturbance frequency ranges, thus increasing the loop gain and reducing the disturbance effects. Since the controller can directly affect only the torque inputs, only the model from control inputs to vehicle rate outputs are required, hence only the A matrix, the control input distribution matrix, and the state to angular rate output distribution matrices are necessary. The system A matrix, however, is still of very high dimension, since it contains all the states resulting from the presence of the disturbance inputs. In order to use the TREETOPS models for design purposes, model reduction techniques have been applied to eliminate these states and reduce the order of the model. Both Schur model reduction and balanced model reduction techniques have been applied and the model has been reduced to as low as 66th order without significantly affecting the input/output frequency responses of interest. The models thus derived, however, are not both stabilizable and detectable, making them unsuitable for controller design with H^∞ design techniques. Reducing the models to orders lower than 66 eliminates significant modes from the system frequency responses, rendering them unusable for design purposes. After extensive effort was extended toward achieving a suitable model for H^∞ design based on the TREETOPS models, it was decided to abandon using these models with H^∞ design algorithms. However, numerical techniques which perform designs based on frequency responses, rather than the models themselves, were successfully used in conjunction with the TREETOPS models. Detailed results of these efforts are presented in a subsequent section.

An apparent inconsistency with the TREETOPS models is the fact the 0° and 90° models are identical. Although the number of disturbance inputs differs between the two models, the input distribution matrices of the models are identical except that one contains columns of zeroes relating to the additional inputs. Thus, from the point of view of frequency response comparisons, the two models are identical. Several of the guest investigators questioned NASA regarding this, but no answer was provided. Whenever these particular models were used, they were assumed to be correct as provided, since no information was available regarding resolution of the inconsistency.

3.3 Non-Composite MIMO Modal Model

Since the models provided in the initial data package were inappropriate for H^∞ controller design techniques, a request to NASA was made to provide a non-composite coupled MIMO modal model of the HST. In late May 1993, Ref. 2 was provided which contains pre-launch modal gain product matrices for 0°, 45°, and 90° solar array orientations. Information is given for 110 modes in each model. Clearly, not all of the modes can be significant, since the composite SISO models only contain modal gain factors for 23 modes. A continuous-time 90° SA orientation model was built by using the modal gain product matrices of the MIMO modal model which correspond to the frequencies of the 23 flexible modes included in the composite SISO modal model. Using this approach and including the rigid body modes, a 52nd order MIMO modal model of the 90° solar array orientation was constructed. Discretizing this model, incorporating a .008 second computational delay modeled as a first degree Padé approximation, and then converting to the w-plane resulted in a model which is minimal, i.e. fully state observable and fully state controllable, allowing it to be used with H^∞ design techniques. Fig. 3.3 presents a representative magnitude response plot of the V_3 axis. It is interesting to note that this plot indicates that the modes present are a subset of the modes of the composite SISO modal model. This is to be expected since the composite SISO model contains modes for all solar array orientations within one model.

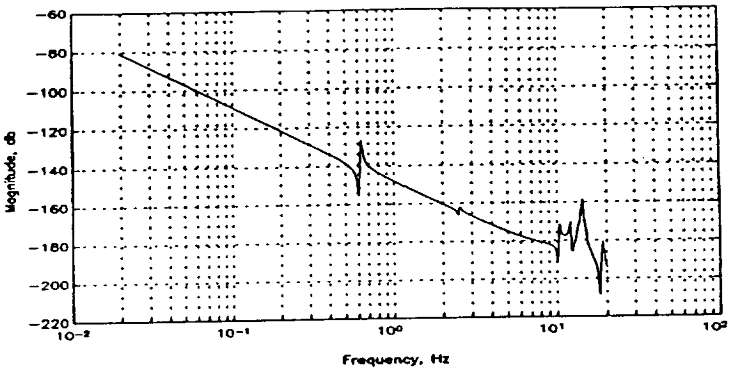

Figure 3.3 Representative (V_3) Non-Composite MIMO Modal Model Magnitude Response

3.4 Design Models vs. Simulation Models

The TREETOPS models are those used by NASA in simulations to evaluate controller performance, while the composite SISO modal models are used for controller design. Due to the fact that most controller design techniques result in controllers of at least the order of the design model, control system design is usually carried out using lower fidelity models than those used in simulation. In the case of the SAGA-II controller design, however, the reverse appears to be true. Examination of the representative magnitude responses presented in Fig. 3.1 and Fig. 3.2 shows that the simulation model (TREETOPS) does not include significant behavior that is known to exhibit itself in the HST plant. It is clear that several modes at roughly 14 Hz do not appear in the TREETOPS model. From information in Ref. 1 these modes appear to be scissors modes and RWA isolator modes.

This lack of fidelity in the simulation models, combined with the SISO nature of the composite SISO modal models, leads to a serious problem in evaluating controllers which appear to give improved performance in these simulations. For example, if a particular controller performs well in the TREETOPS based simulation, there is no guarantee, or any reliable indication, that the controller would even stabilize the actual system or stabilize a higher fidelity simulation. In light of these considerations, it was decided that simulations would be performed with both the TREETOPS based models and the non-composite MIMO modal models in order to obtain a more thorough evaluation of controller performance. Further details of the simulation implementation are presented in a subsequent section.

4.0 SIMULATION

Computer simulations are essential tools in the evaluation of control system performance. Two simulations of the HST pointing control system are provided by NASA for the redesign effort; a linear MATLAB® simulation and a nonlinear simulation written in Microsoft® Fortran. Both simulations are based on a method whereby actual recorded in-flight vehicle rate data is "played back" through the simulation, yielding the vehicle rate data which would result if the controller being simulated had been in the HST rather than the controller in place when the flight data was recorded. This assumes, of course, that the plant model in the simulation exactly emulates the real HST plant. Also, in order for such a simulation to be meaningful, it is assumed that the disturbance torques present when the flight data was recorded are typical disturbances. Regardless, if the plant model is an accurate representation of the true HST plant, the simulation allows comparison of the performance of a redesigned controller to that of the in-flight controller under the application of an identical set of disturbance inputs. This section reviews the derivation of the simulation, presents the steps taken to achieve successful application of the MATLAB® linear simulation to controller evaluation, and presents the modifications made to the Fortran simulation to make it useful in the redesign effort. In particular, the modifications necessary to provide a realistic simulation of fixed point arithmetic for the redesigned controllers are discussed.

4.1 Simulation Derivation

Ref. 1 contains a derivation of the simulation methods based on a SISO system, but the simulation is actually applied to a MIMO system. A MIMO derivation is presented to illustrate the theoretical validity of the MIMO simulation.

Fig. 4.1 displays a simplified single loop control system which typifies the case for which a disturbance input affects a plant output.

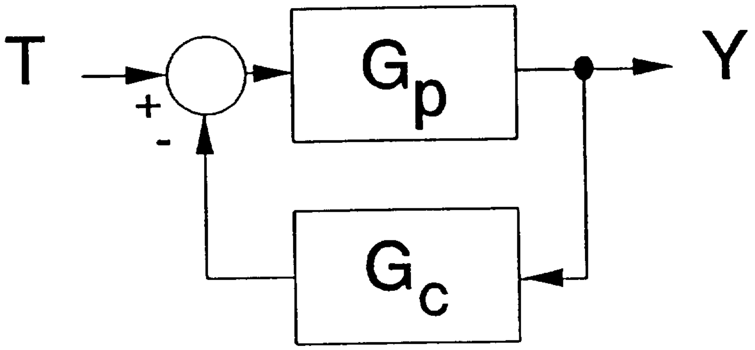

Figure 4.1 Single Loop Control System

Writing equations for the output yields

$$Y=G_p(T-G_cY), \tag{4.1}$$

which, when rearranged, becomes

$$Y=(I+G_pG_c)^{-1}G_pT. \tag{4.2}$$

Equation 4.1 describes the vehicle rate output due to a disturbance torque input in vehicle coordinates for the HST if G_p is considered to be the HST plant and G_c to be the HST controller.

Fig 4.2 shows another block diagram, which is useful in illustrating how the recorded flight data can be "played back" through the simulation to yield the HST response to a set of disturbances with a different controller in place.

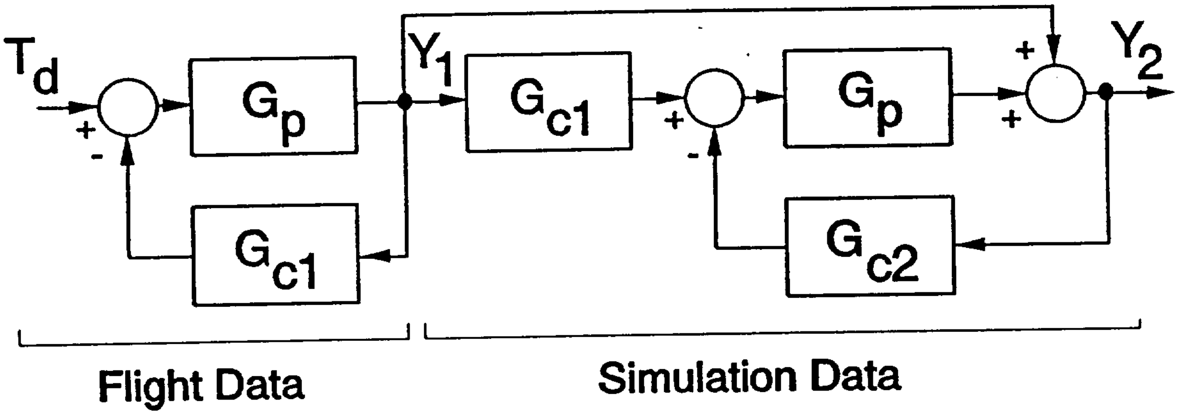

Figure 4.2 HST Simulation Block Diagram

The left part of the diagram (labeled Flight Data) can be considered to be the HST, where G_{c1} is the PID controller, and Y_1 is the recorded flight data. The right portion of the diagram (labeled Simulation Data) is the block diagram which represents the computer simulation under which the redesigned controllers are evaluated. The equation relating Y_1 to T_d is

$$Y_1=(I+G_{c1}G_p)^{-1}G_pT_d. \tag{4.3}$$

Now, the equation for Y_2 in terms of Y_1 is

$$Y_2=Y_1+G_p(G_{c1}Y_1-G_{c2}Y_2), \tag{4.4}$$

which, rearranged with terms of Y_2 on the left and terms of Y_1 on the right, becomes

$$(I+G_pG_{c2})Y_2=(I+G_pG_{cl})Y_1. \tag{4.5}$$

Then

$$Y_2=(I+G_pG_{c2})^{-1}(I+G_pG_{cl})Y_1, \tag{4.6}$$

and inserting the expression for Y_1 in terms of T_d yields

$$Y_2=(I+G_pG_{c2})^{-1}(I+G_pG_{cl})(I+G_pG_{cl})^{-1}G_pT_d, \tag{4.7}$$

or

$$Y_2=(I+G_pG_{cl})^{-1}G_pT_d. \tag{4.8}$$

Equation 4.8 is the equation that would result from Fig. 4.1 if G_{c2} were the in-flight controller. Thus, the derivation has shown that the simulation block diagram of Fig. 4.2 yields the response of the HST with G_{c2} as the in-flight controller to the set of torque disturbances present while the flight data was taken. It should be reiterated that this is true only under the assumption that the simulation plant model perfectly models the actual HST plant.

4.2 Linear Simulation

A linear simulation was provided as a MATLAB® program with the original project information packet. However, the program would not correctly load the flight data files which were provided as the simulation input. After some time conferring with NASA about the flight data files, they were altered to correctly load into the MATLAB® program, and the linear simulation was successfully implemented and used as a preliminary evaluation of controller performance. Since only 0° and 90° degree flight data were provided to drive the simulation, but the 0° and 90° TREETOPS plant models utilized by the simulation were identical, the simulation input data was examined to see which exhibited the largest LOS errors when simulating the SAGA-II controller. That case only was used as the basis for comparison among redesigned controllers. All simulations subsequently performed use the 90° flight data, since it appears to be the "worst case" available data. Any controller's performance can be examined in terms of some baseline controller, but no absolute performance measures can be claimed from results of these simulations. Since SAGA-II was chosen to be the controller against which any new designs were compared, all simulation results are presented in comparison with SAGA-II simulation results with identical input data. No simulations for 45° or 105° solar array orientations were performed since appropriate simulation input data was not provided.

4.3 Fortran Simulation

Another simulation written in Microsoft® Fortran was provided by NASA. This is a nonlinear simulation which appears to account for some fixed point arithmetic effects by including the software limiters of the controller implementations in the DF 224 flight computer. The simulation also includes the 0.8 N-m torque limits of the RWA units. The MATLAB® simulation does not provide for these effects in any way, nor would it be a simple matter to include them, so an effort was undertaken to use the Fortran simulation for more meaningful evaluation of controller designs.

The Fortran simulation, as provided, has several drawbacks. The drawbacks, along with the steps taken to overcome them, are:

(1) The Fortran simulation was constrained to run on a DOS computer since it was written in Microsoft® Fortran. While this is not in itself a serious problem, it was found that simulations dealing with input flight data recorded over hundreds of seconds took nearly a half hour of computer time to run. It was felt that a speedier simulation was necessary, so an effort was undertaken to implement the Fortran simulation on a Sun Sparc® workstation. This required compiling the simulation under the Sun Sparc® version of Fortran 77. The simulation loads the input data and the plant model from MATLAB® .mat files, which are pure binary files. While Microsoft® Fortran can directly read such a file, the available version of Fortran 77 could not. In order to overcome this problem, programs were written in C which allow Fortran 77 to read binary files through calls to the C routines. The modified simulation executes roughly 6 times faster than the DOS version.

(2) The original Fortran simulation is hard coded such that the G_{c2} block of Fig. 4.2 must be in the form of the existing HST controllers. Thus the original simulation is useless in the study of controllers which are anything more than just slight changes to the existing PID, SAGA-I or SAGA-II controllers. The modified simulation implements G_{c2} as a state space realization, so any controller may be evaluated, regardless of its structure.

(3) The PID data file provided with the simulation contained errors, but no information was available regarding their correction. It appears the data files which accompanied the simulation were files generated during a sensitivity study, but it is not known what the file for the actual in-flight PID controller should be. Since no complete information detailing the contents of the data file was available, but an alternative state space model of the PID controller was on hand, a decision was made to implement the G_{c1} controller block of Fig. 4.2 as a state space realization.

(4) The plant model provided with the Fortran simulation failed to take into account

the vehicle inertia matrix and a differentiator present in the PCS control loop, with the result that the PID and SAGA-II controllers did not stabilize the simulation plant model. A new simulation plant model based on the TREETOPS 90° model which includes the inertia matrix and the differentiator was built to solve this problem. An alternative plant model based on the 90° non-composite MIMO modal model was also constructed. The PID and SAGA-II controllers stabilize both new simulation plant models.

(5) The original simulation appears to use software limiters to simulate the effects of fixed point arithmetic, whereas more accurate methods are available. Section 4.4 details the implementation in the modified simulation which uses better methods to provide a more accurate simulation of fixed point arithmetic effects.

As the modified Fortran simulation was developed, the G_{c1} and G_{c2} controller blocks of Fig. 4.2 were implemented as state space realizations for the reasons listed in items (2) and (3) above. The simulation was programmed so that the investigator can select torque limits and/or fixed point arithmetic effects at run time. If the torque limits and fixed point effects are not selected, a linear simulation is performed. The Fortran simulation is also useful because one can "get inside" and examine signals internal to the simulation block diagram (Fig. 4.3) or constrain them to perform sensitivity analyses. This was impossible with the MATLAB® simulation. After verifying that the Fortran linear simulation yields identical results to the MATLAB® linear simulation for an identical simulation run, the MATLAB® simulation was no longer used since the Fortran simulation executed much more quickly.

4.4 Fixed Point Arithmetic Simulation

For most digital control systems or digital signal processing systems, it is necessary to determine the performance of the system through comprehensive simulation before actual implementation. In the case at hand, it is required that a comprehensive simulation be performed to insure that the re-designed control system will perform in an acceptable manner given the limited computational capabilities of the flight computer of the HST pointing control system controller.

In the development of a comprehensive system simulation, a problem arises from the fact that the computer performing the simulation typically has a longer wordlength than that of the microprocessor which is to be used in the actual implementation. This introduces the difficulty of forcing the computer performing the simulation to generate the same results as the shorter wordlength microprocessor. The following paragraphs contain a description of one method of performing fixed-point arithmetic on a computer which utilizes floating point notation.

The flight computer presently in use on the HST inherently represents numbers as integers. However, it appears the user is free to assume a radix point anywhere in the binary word. Obviously it is possible to use floating point arithmetic (as is used on most microcomputers, minicomputers and mainframes) through additional programming effort to decrease quantization errors, but floating point arithmetic increases the execution time. Since execution time is a major concern in implementing real time control systems with a microprocessor, fixed point arithmetic is desirable, assuming sufficient accuracy is available.

On the other hand, simulation of large systems is typically performed on computers which use a larger wordlength than used in the HST flight computer. Thus, in order to accurately simulate the pointing control system in a typical simulation, special coding for performing the fixed-point computations of the flight controller should be included.

The special coding which must be included in the simulation to perform microprocessor type arithmetic must take into consideration three sources of errors which are caused by fixed point calculations. First, the coefficients of the desired digital signal processing algorithm may not be exactly representable by the microprocessor. Secondly, any inputs or outputs of the system may not be converted precisely due to the finite wordlengths of the A/D's and D/A's. Thirdly, any arithmetic operations performed by the microprocessor (e.g., additions or multiplications) are subject to quantization to the nearest representable result.

All three errors mentioned stem from the same basic cause, the finite wordlength of the system components. These errors must all be taken into account when writing a comprehensive simulation code which can be used to determine system performance. The code must truncate or round off numbers at each appropriate time in the calculations in order to correctly simulate the system. **The basic problem to be solved in designing the portion of the simulation code which computes the response of the digital portion of the system is to write a code which performs all pointing control system calculations in such a manner as to produce the same results obtained using a fixed point processor.** It is also necessary in general to be able perform these calculations with any specified wordlength in order to simulate microprocessors of different wordlengths. In this case, since the DF-224 appears to be a 24 bit processor with 13 bits to the right of the radix, this quantization level is used in the simulation. The user is allowed to change this quantization scheme if desired, e.g. for consideration of alternative processors.

There are two basic approaches which may be used to implement the fixed-point arithmetic in a general simulation. The first approach uses a set of inline functions to produce results expected from each of the operations generally associated with fixed-point arithmetic control systems. These include the A/D process, multiplication, addition (or subtraction), change of radix position or wordlength, and the D/A process. In this case, the floating point numbers used at each step of the simulation are manipulated in such a way as to produce the integer results which would be anticipated at each stage of the actual system calculations. These integers are available for inspection at each stage of the calculations, and must be converted back to a decimal format (by means of the D/A process) before being fed into the continuous system simulation.

Another approach reduces the programming requirement to a single inline function which quantizes a given number according to a given wordlength and radix position, but returns the decimal representation of the number rather than the integer representation. This technique significantly reduces the complexity of the simulation in terms of the number of different function calls required. However, additional computation overhead is incurred by converting the decimal representation to an integer representation at each step of the simulation. This tradeoff of speed versus simplicity must be considered in large system simulations, but speed is not considered to be a significant problem for this specific problem since a limited number of simulations will be performed.

The actual calculations can be performed in an inline function "quant(x,info)", where "x" is the number to be operated on, and the vector "info" contains the quantization details. The actual implementation of function "quant" is a two-step process in which the number is first converted to an equivalent integer representation of the fixed-point number by means of

$$N(X) = IP[X \cdot 2^{p} + r \cdot sgn(X)]. \tag{4.9}$$

In Equation 4.9, N(X) is the resultant integer, IP[ ] is the integer part operator, X is the number to be operated on, p is the number of binary places desired to the right of the radix point, r is the designation for rounding or truncation (0 for truncation and 0.5 for rounding), and sgn( ) is the sign operator. Following this conversion to integer format, the number is compared to the largest number representable, given as

$$Z = 2^{i} - 1. \tag{4.10}$$

Z represents the maximum number and i represents the number of bits in a single fixed point word. It should be noted that one bit of the overall wordlength is typically required as a sign bit, so for the 24-bit processor in the HST pointing control system, only 23 are apparently available for magnitude representation. Finally, a return to decimal representation is by means of

$$D(X) = \frac{X}{2^{p}} . \tag{4.11}$$

In equation 4.11, D(X) represents the resultant decimal number, X is again the number to be operated on, and p is again the number of binary places desired to the right of the radix point.

For example, if the number 6.37 is to be represented using a rounded 8-bit integer format with 2 bits to the right of the radix, Equation 4.1 yields:

$$N \;=\; IP[(6.37)(2^2) + 0.5]$$

or

$$N = 25.$$

This represents the binary number $(11001)_2$. Note that when the radix point is inserted in the desired position, (or, equivalently, Equation 4.3 is applied) the number is $(110.01)_2 = (6.25)_{10}$, which is the decimal representation of the quantized number. Also note that for the case in which one bit is reserved for a sign bit, the maximum permissible number is $2^7-1 = 127$. This is larger than the value 25 calculated above, and it is seen that no overflow occurs.

This method of massaging decimal numbers provides the basis for the simulation of the fixed-point arithmetic used in the modified HST nonlinear simulation. In order to accurately simulate the microprocessor-realized arithmetic, each operation required by the processing algorithm must be performed in an appropriate manner. This entails an initial quantization of any inputs and any coefficients to be used in the controller implementation, as well as quantization of each arithmetic operation.

The modified linear Fortran simulation program required modifications to two specific subroutines, those corresponding to the G_{c1} and G_{c2} simulation blocks, in order to implement fixed-point arithmetic for the digital controller calculations. The modifications are (1) to quantize each coefficient of the controllers, (2) quantize inputs to the controllers, and (3) quantize any calculations performed by the fixed-point processor.

Each of these controllers is implemented as a state space realization in the modified Fortran linear simulation. Thus, the coefficient quantization is a simple matter of quantizing each element of the A, B, C, and D matrices in each routine. The quantization of the input vectors follows in a similar fashion. When running the simulation under the assumption of a 24 bit processor with 13 bits to the right of the radix point, a slight difficulty is encountered in that the G_{c2} controller calculation has the problem of underflow of the input terms. For one particular run, the G_{c2} inputs were observed to be on the order of 10^{-9} for the entire simulation, which is considerably smaller than 1 LSB for the given radix position. (1 LSB for 13 bits to the right of the radix is $2^{-13} = 122.0703125 \times 10^{-6}$.) Thus, the G_{c2} controller received all zero input for all simulated time. For the purposes of this simulation, a scale factor of 10^6 is used at the input to the G_{c2} controller, and an inverse scaling by a factor of 10^{-6} is used at the output. The G_{c1} controller experiences no such difficulties.

The final required quantization involves the arithmetic operations used in the calculation of the next states and the current outputs. A simple inline quantization at each step of the calculation including each multiplication and addition in the required matrix calculations for the state equation computations is all that is necessary to complete the quantization process.

A final comment regarding the specific implementation is that, even though the actual system appears to use a 24-bit processor with 13 places assumed to the right of the radix, the simulation was not "hard-wired" for this type quantization. Each of the two digital controllers is individually quantizable, giving the system engineer the capability to study the quantization effects of the processors alone or in combination. This flexibility was in fact useful in determining that the underflow problems were originating in the G_{c2} controller by checking each controller separately. Quantization of the G_{c1} controller did not result in zero output, but quantization of the G_{c2} controller did. Additionally, this allows for studies of implementation requirements in the case of computational hardware upgrades.

While it has not been possible to ascertain the precise details of the DF-224 controller algorithm, the fixed point implementation included in the modified simulation provides the only available means to investigate the effects of fixed point arithmetic on the performance of the redesigned controllers. It is also important to note that once the structure of the algorithm in the DF-224 is known, these details can be emulated by a judicious choice of the state space realizations of G_{c1} and G_{c2}. The experience of the authors is that emulating the structure of the control processor algorithms is critical; i.e., the closed loop performance of a controller implemented in fixed point arithmetic can be highly sensitive to the algorithm structure.[3]

5.0 H^∞ CONTROLLER DESIGN

H^∞ controller design is a MIMO loop shaping approach which determines a controller which satisfies desired performance and robustness specifications. The HST redesign problem provides an excellent opportunity to apply H^∞ design techniques to a complex "real world" MIMO system. Such an exercise allows the comparison of H^∞ controller design to other design methods, and yields an indication of the practicality of using H^∞ techniques on complex realistic problems. The authors have previously applied analytical H^∞ design theory to several problems (Ref. 4 and Ref. 5). This section outlines the application of H^∞ design techniques to the HST redesign.

5.1 Background and Design Philosophy

The philosophy applied to the HST pointing control system redesign is to obtain high controller gain in the frequency range of the dominant disturbances, thereby accomplishing some degree of disturbance rejection. The loop shaping approach of H^∞ controller design theory is ideally suited to achieving this type of specification. In using H^∞ design algorithms, frequency dependent weighting functions are applied to certain outputs of the control system; these weighting functions specify the desired system performance and robustness.

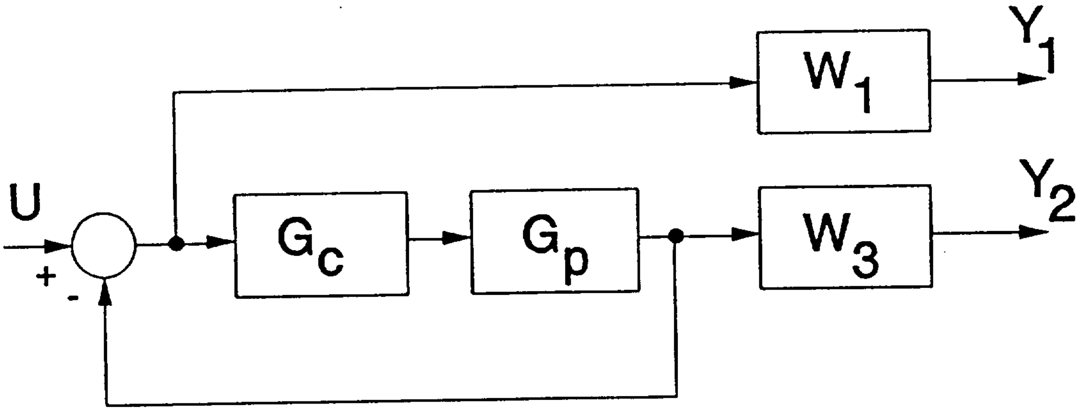

Figure 5.1 Block Diagram for H^∞ Controller Design

Fig. 5.1 shows the block diagram used to set up a typical H^∞ design problem. The weighting function W_1 is applied to the error signal, and if this signal is denoted as E, it is apparent that $E=SU$ where $S=(I+G_pG_c)^{-1}$, which is commonly known as the sensitivity function. If the plant output is denoted as V, it is clear that W_3 weights the function $V=TU$, where $T=(I+G_pG_c)^{-1}G_pG_c$, which is known as the complementary sensitivity function. The use of W_1 and W_3 is referred to as the mixed sensitivity approach to H^∞ design. Once W_1 and W_3 are specified, the H^∞ design algorithm determines whether a controller exists which satisfies the constraints that (1) the frequency dependent singular values of the sensitivity function lie below the inverse of the W_1 weighting function and (2) the singular values of the complementary sensitivity function lie below the inverse of the W_3 weighting function. Thus, H^∞ design is a loop shaping process. H^∞ is iterative in nature in the sense that there is no guarantee that the specifications imposed by a given set of weighting criteria can be met. If no controller exists which satisfies the specifications, the specifications must be relaxed until a controller can be found which satisfies them. The reader interested in further details of H^∞ design philosophy and algorithms is referred to the References 4, 6, 7, and 8, and the references therein.

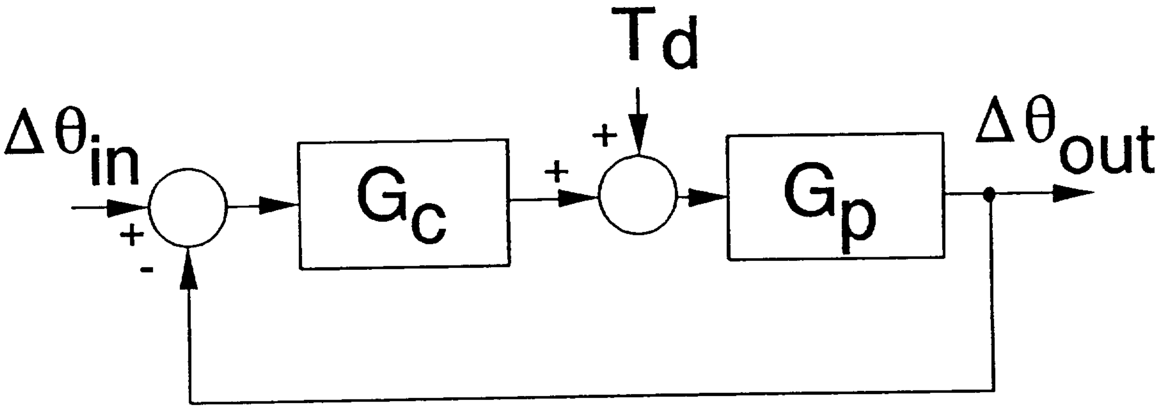

Figure 5.2 HST Pointing Control System Disturbance Model

Fig. 5.2 shows the simplified disturbance model for the HST PCS system. Writing the equation for the contribution to the output by the disturbance yields $\Delta\theta_{out}=(I+G_pG_c)^{-1}G_pT_d$, or $\Delta\theta_{out}=SG_pT_d$, where $S=(I+G_pG_c)^{-1}$, the sensitivity function previously described. Since the controller can be designed for attenuation of the sensitivity function in the disturbance frequency band, it is easy to see how sensitivity function design can provide some degree of rejection of disturbances in the system output. On the other hand, providing attenuation of the complementary sensitivity function provides robustness to model uncertainty in the frequency range of the attenuation. It should be noted, however, that the designer cannot "have it all"; attenuation of either sensitivity or complementary sensitivity may be achievable in a given frequency range, but not both. The philosophy applied to the HST redesign is (1) to achieve sensitivity function attenuation in the low frequency ranges where the solar array disturbances are known to exist, thereby reducing the effects of the disturbances, and (2) to achieve as much robustness to model uncertainty as possible while meeting the disturbance attenuation specifications.

5.2 H^∞ Design Problem Setup for the HST

The HST controller must contain an integrator in each axis in order to retain the number of free integrators required to eliminate steady state position errors caused by gravity gradients and aerodynamic forces. In addition, any controllers designed must be 42nd order or less so that they can be implemented in the DF-224 flight computer. While maintaining a given controller order when using standard H^∞ design algorithms with high order plants is difficult, if not impossible, controller order can be reduced by using reduced order plant models when performing the design and then possibly applying order reduction techniques to the resulting controller. It is possible to meet the integrator term constraint through careful setup of the design problem.

Since the SAGA-II controller is essentially a PID controller in cascade with some notch filters, a similar approach was followed in the H^∞ design; i.e., the overall HST controller would be the original PID controller (without the FIR filters for the RWA isolators) in cascade with the controller achieved through H^∞ design techniques. Fig. 5.3 shows the block diagram of such an implementation.

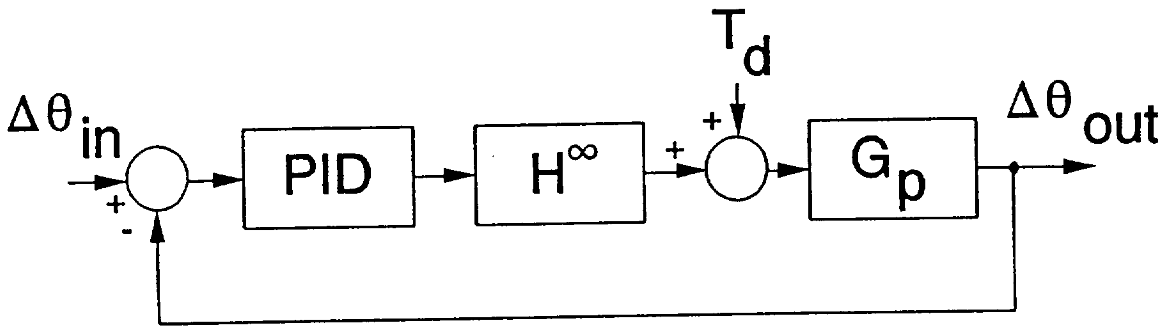

Figure 5.3 HST H^∞ Controller Implementation

Placing the H^∞ controller after the PID controller maintains system type, since the PID controller contains the required integrators.

Fig. 5.1 shows the form into which a control system must be cast in order to apply the H^∞ design algorithm. Everything outside of the H^∞ controller is seen by the algorithm as part of the plant. Clearly Fig. 5.3 is not in the required form for H^∞ design, so some rearrangement must be undertaken before attempting to use the H^∞ software. If the following manipulations are performed on Fig. 5.3, the block diagram of Fig. 5.4 results.

(1) Assume T_d is zero.

(2) Sum an input U after the PID and before the H^∞ controller.

(3) Multiply the PID output by -1; denote this signal as Y.

(4) Weight Y by W_3.

(5) Sum the negative of Y with U at the input to the H^∞ controller.

(6) Weight the input signal to the H^∞ controller by W_1.

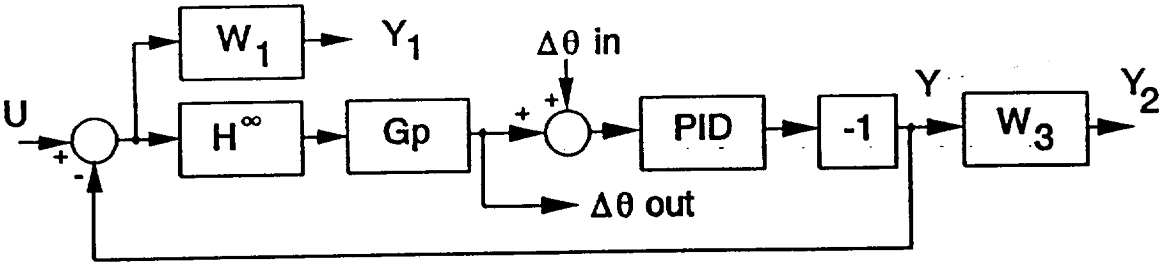

Figure 5.4 Rearranged System with Weighting Functions

Observe that if: (1) U is zero, (2) the weighting functions are unity, and (3) a disturbance input summed in just prior to G_p, Fig. 5.4 becomes the same block diagram as Fig. 5.3. Thus, in obtaining Fig. 5.4 from Fig. 5.3, all that has been done is some rearranging, augmentation by weighting functions, and addition of inputs. Any controller which stabilizes the closed loop of Fig. 5.4 and provides high loop gain will do so regardless of the number of inputs or their points of application. Thus, it is valid to design with this block diagram to achieve the desired results when the controller is implemented as in Fig. 5.3. Using Fig. 5.4, where everything between the H^∞ block and the W_3 block is considered to be the plant, for design purposes, the HST PCS has been put into form of Fig. 5.1, the setup required for H^∞ design.

H^∞ design algorithms cannot achieve successful controller designs if the augmented plant contains poles on the imaginary axis. Since the augmented plant of Fig. 5.4 contains rigid body modes on the imaginary axis, some sort of plant modification is necessary. Often alpha shifting is used, but unless carefully applied, stabilizing an alpha shifted plant in the presence of W_1 and W_3 doesn't guarantee stabilization of the unshifted plant. Rather than alpha shift the plant, a method was employed whereby a constant "prefeedback" was applied to the plant as seen by the H^∞ controller to move the rigid body modes off of the imaginary axis. Fig. 5.5 illustrates the block diagram of the problem setup for input to the H^∞ algorithm with prefeedback in place. The prefeedback is then included as part of the actual implemented controller, as illustrated in Fig. 5.6, with the result that the plant with prefeedback which is stabilized by the H^∞ controller is the plant that actually exists when the design is complete, since the prefeedback is included in the controller. Referring to Fig. 5.6, it can be observed that the prefeedback does not wrap around the PID controller, so it cannot cause any reduction in the low frequency gain of the system. As a result, the system type is maintained.

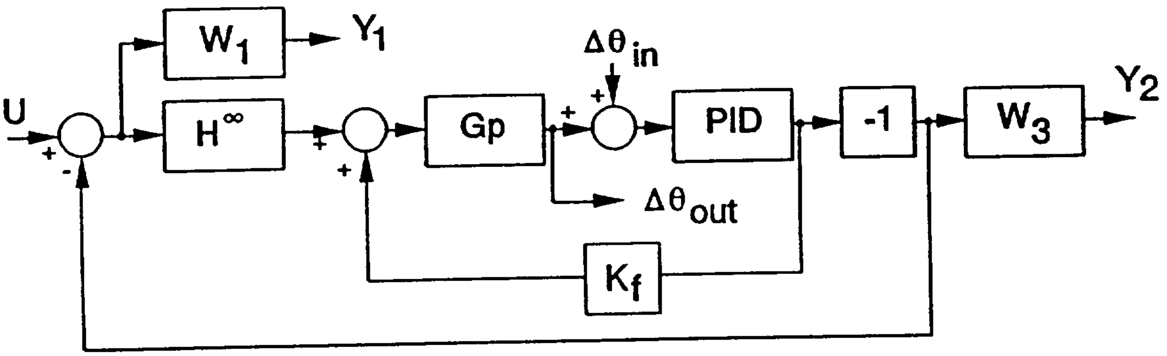

Figure 5.5 H$^\infty$ Design Problem Setup with Prefeedback

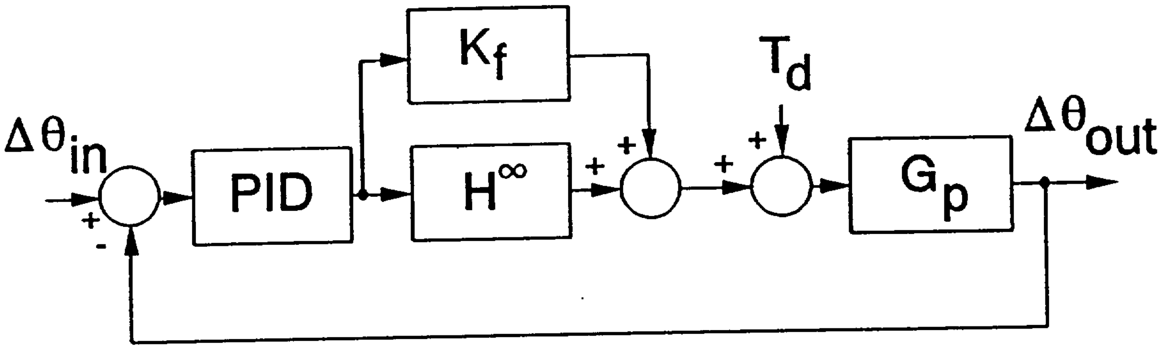

Figure 5.6 H$^\infty$ Controller with Prefeedback Included

Using the problem setup shown in Fig. 5.5, initial attempts at controller design were performed based on the TREETOPS models for G_p. Since the TREETOPS models are over 160th order, balanced model reduction and Schur model reduction techniques were applied to reduce the models to 66th order. Any further reduction results in the loss of significant modes from the frequency responses of the models. Since the TREETOPS models contain rigid body modes, it is necessary to alpha shift them slightly (by + or - .00001 Hz or more) prior to applying reduction techniques since the reduction algorithms cannot deal with rigid body modes. It was found that regardless of the alpha shift, the resulting reduced models were not stabilizable and/or not detectable, rendering them useless for H$^\infty$ design. The full order TREETOPS models exhibited the same behavior. After expending considerable effort to remedy these problems, H$^\infty$ design attempts based on the TREETOPS plant models were abandoned. Other types of design techniques were successfully applied to these models, however, and are presented in a subsequent section.

It is felt that the composite SISO modal models are not valid models for H$^\infty$ design for two reasons: (1) they are composite in nature, i.e. there is information about more than one solar array orientation contained in the same model, and (2) they are SISO models and insufficient information was provided to correctly couple the SISO models into a MIMO model. Since H^∞ is a true MIMO design technique, using SISO plant models would not be a valid illustration of the MIMO capabilities of H^∞.

After requests to NASA for a non-composite MIMO model, such a model was provided[2] as a set of modal gain product matrices in late May, 1993. A MIMO modal plant model was constructed as described in Section 3, and the H^∞ controller design efforts were renewed.

5.3 H^∞ Controller Design Results

Setting up the problem as shown in Fig. 5.5, and using the 90° non-composite MIMO modal plant model, successful H^∞ controller designs were achieved. The designs were successful in the sense that, under linear analysis and simulation, they stabilized the modal plant model and provided improved disturbance attenuation over the SAGA-II controller implementation. The results of the H^∞ controller design which yielded the greatest performance improvement over SAGA-II are presented.

A part of setting up the problem for H^∞ controller design is the specification of the prefeedback matrix and the weighting functions. The prefeedback must be selected such that the rigid body mode poles are moved off of the imaginary axis, but no other poles are moved onto the imaginary axis or so far into the right half plane that they cannot be stabilized. Furthermore, prefeedback gains greater than unity effectively increase the gain of the PID controller, increasing loop gains at low frequencies. The H^∞ controller must then stabilize the system which contains the prefeedback and meets the specifications indicated by the weighting functions. A prefeedback matrix meeting these criteria and subsequently used in the H^∞ design attempts is

$$K_f = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

The modified plant in Fig. 5.5, consisting of the blocks G_p, PID, -1, and K_f was built in the z-plane and then converted to a w-plane realization, since the H^∞ design algorithms must be applied to continuous systems. The weighting functions were then created to obtain high controller gain in the low frequency ranges where the disturbance energy generated by the solar arrays are known to exist.

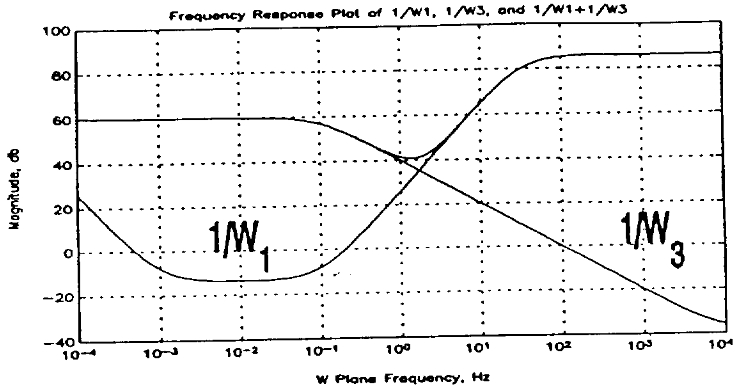

Figure 5.7 Weighting Functions for H^∞ Design

Fig. 5.7 displays the w-plane weighting functions used in designing the controller. The W_1 weighting function specifies 14 db of attenuation in the w-plane frequency range of .001 to .1 Hz. Additional attenuation is provided by the fact that the K_f gain matrix amplifies the gain of the PID controller; i.e. if the H^∞ controller were not in place, examination of Fig. 5.6 reveals that the loop gain would be boosted by the presence of the K_f block alone. The function of the H^∞ controller is to stabilize this amplified loop, and also provide the singular value characteristics specified by the weighting functions. W_3 specifies robustness to model uncertainty at w-plane frequencies above 100 Hz.

The design algorithms were used to obtain an 82nd order H^∞ controller which meets the above specifiations. Since the augmented plant is 82nd order, this is the smallest controller order which can be obtained. The controller was inserted into the pointing control system loop as shown in Fig. 5.6 and compared with the SAGA-II implementation via frequency domain analysis and time domain simulation. All comparisons were performed with the 90° non-composite MIMO modal plant model, since that is the model with which the H^∞ design was performed. Hence, all comparisons with SAGA-II use the same plant model in order to maintain consistency. The disturbance rejection achieved with the H^∞ controller is superior to that of SAGA-II in all three vehicle axes. As a typical example, Fig. 5.8 shows the magnitude of the disturbance to output frequency response for the V_3 HST axis with the H^∞ controller, and Fig. 5.9 displays the same response with the SAGA-II controller.

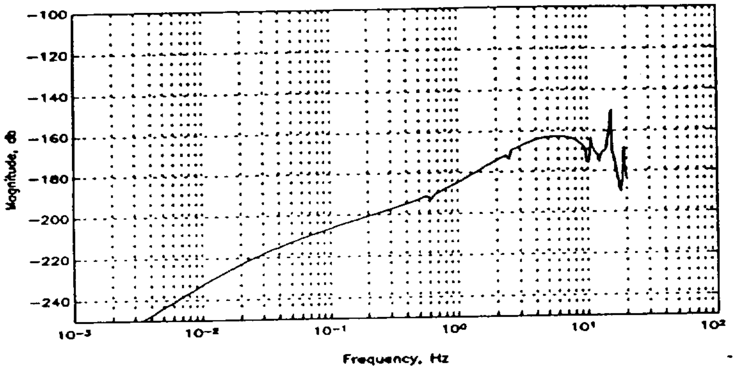

Figure 5.8 V_3 Disturbance to Output Magnitude Frequency Response with H^∞ Controller

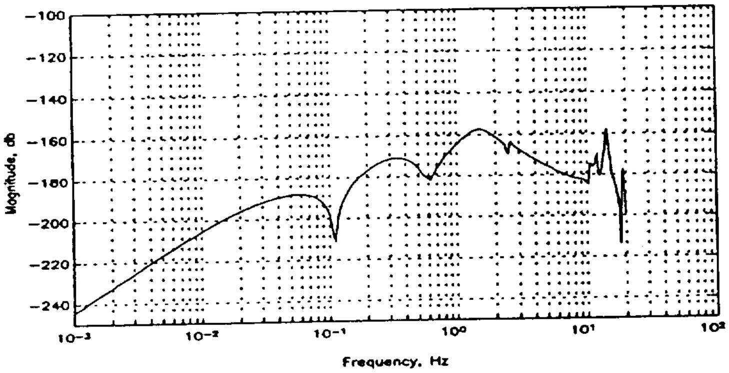

Figure 5.9 V_3 Disturbance to Output Magnitude Frequency Response with SAGA-II Controller

Comparison of the two figures reveals that the disturbance to output frequency response of the HST PCS with the H^∞ controller has up to 20 dB greater attenuation of disturbances than the SAGA-II case at all frequencies below 2 Hz, but at frequencies above 2 Hz the attenuation is greater for SAGA-II. Thus, the design goal of increased low frequency disturbance attenuation over the SAGA-II controller was attained. Similar results were achieved in the other two axes.

Since the prevailing opinion is that the disturbances are primarily low frequency in nature, it was felt that the design would provide superior performance to SAGA-II because of the broad band low frequency attenuation. In order to verify the performance of the new controller, several simulations were run. The first simulation was a linear simulation, without torque limits, of the system with the H^∞ controller. The results of the simulation for the V_3 LOS are presented in Fig. 5.10. Figure 5.11 shows the results of the same simulation applied to the SAGA-II controller.

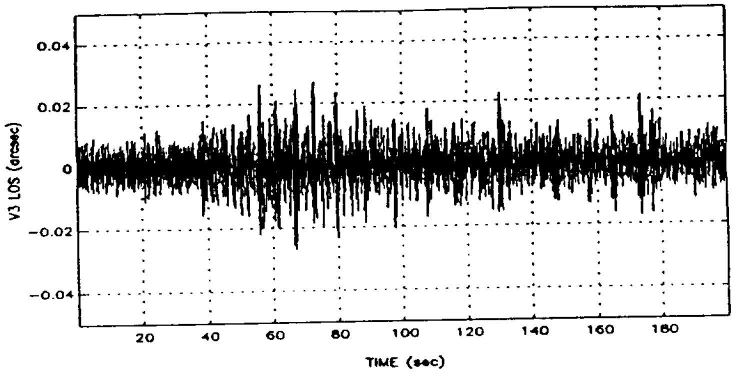

Figure 5.10 Simulated V_3 LOS with H^∞ Controller

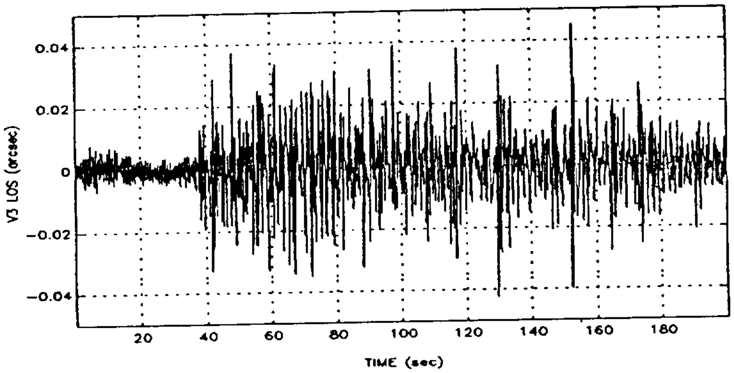

Figure 5.11 Simulated V_3 LOS with SAGA-II Controller

Table 5.1 contains a numerical comparison of the simulated LOS data of the two controllers for all three vehicle axes. The tabulated results are based on 500 seconds of on-orbit data including a day-to-night terminator.

Table 5.1 Peak and RMS Attitude Values for Linear Simulation with Modal Model

HST Axis		V_1	V_2	V_3
Peak Value (milliarcsec)	SAGA II	136.8	28.9	67.8
	H^∞	70.2	12.3	48.3
RMS Value (milliarcsec)	SAGA II	26.4	3.44	6.67
	H^∞	9.71	2.44	4.63

Figures 5.12 and 5.13 display the power spectral density (PSD) obtained from the V_3 LOS data for the H^∞ and the SAGA-II controllers, respectively.

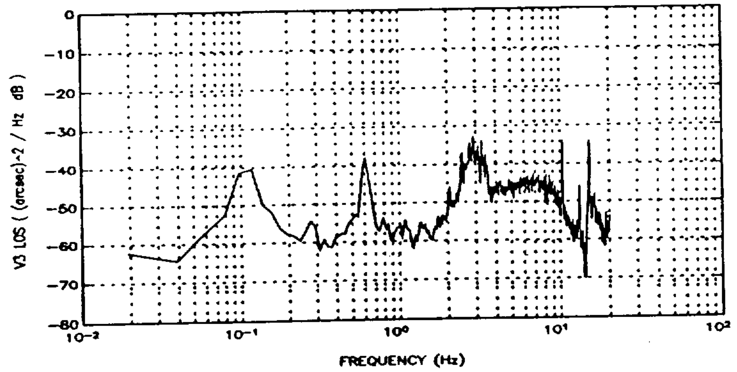

Figure 5.12 V_3 PSD Estimate from Linear Simulation with H^∞ Controller and No Torque Limits

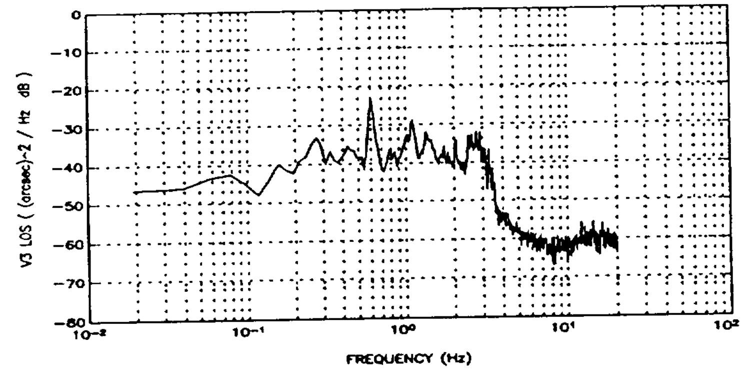

Figure 5.13 V_3 PSD Estimate from Linear Simulation with SAGA-II Controller and No Torque Limits

The H^∞ controller reduces the PSD of the V_3 LOS data in the low frequency ranges, but increases it in the higher frequencies. This is as expected, since the disturbance to output magnitude frequency responses in Figs. 5.8 and 5.9 show that the H^∞ controller causes greater low frequency attenuation but less attenuation in the high frequencies than the SAGA-II case. In the case of the HST PCS, this results in improved performance since the solar array disturbances occur at low frequencies. Again, similar results were obtained in the other two axes.

Clearly, the H^∞ controller provides improved performance over the SAGA-II controller in all three vehicle axes in the linear simulation with no torque limits, using the 90° non-composite MIMO modal plant model. Broadband low frequency disturbance attenuation proves effective in reducing the disturbance effects.

In order to further examine controller performance, the linear simulation with torque limits in place was performed, with the result that the inclusion of torque limits destabilizes the system. Examination of the linear simulation without torque limits reveals that the torque signals of the system with the H^∞ controller in place are as high as 40 N-m, so it is understandable that limiting the torques to .8 N-m destabilizes the system.

In order to test the robustness of the controller to changes in the plant model, another simulation was performed which omitted the torque limits, but used a TREETOPS plant model rather than the 90° non-composite MIMO modal plant model, with the result that again the system proves to be unstable. This is especially interesting, since the frequency response of the TREETOPS plant models appears to contain fewer modes than the MIMO modal plant model. It was subsequently found that the H^∞ controller does not stabilize any available HST model except the model for which it was designed.

Model reduction techniques were applied to the 82nd order H^∞ controller in order to reduce it below 42nd order. No reduced order controllers stabilize the plant model for which the original H^∞ controller was obtained, nor any other available HST plant model. All simulation attempts with these reduced order controllers indicate instability.

5.4 H^∞ Summary and Conclusions

The H^∞ controller design was a complicated and time consuming process. Using the 90° non-composite MIMO modal HST plant model, an 82nd H^∞ order controller was developed. The controller provides superior simulation performance to the SAGA-II controller when linear simulations are run with the plant model used in the H^∞ design. However, the controller destabilizes the system when torque limits are included, or when any HST plant model other than the one for which the H^∞ controller was designed are used as the simulation plant. Furthermore, reduced order versions of the H^∞ controller fail to stabilize any available HST plant model. Clearly, the controller obtained from the H^∞ design algorithms is not, in reality, a controller which could be flown in the HST. Despite the fact that a controller was designed which improves over the SAGA-II controller in some respects, the practicality of standard H^∞ design techniques applied to a system as complex as the HST must be questioned.

6.0 NUMERICAL REDESIGN

In order to overcome the numerical difficulties that were encountered with the standard state-space approach to designing controllers for achieving closed loop H^∞ design specifications and to obtain a controller of an acceptable order, an iterative numerical method was employed for design. This method, which is similar to multiple objective optimization, has been previously used to aid in the design of an effective controller for a large space structure ground test facility at NASA Marshall Space Flight Center[9]. A few of the advantages of this method over analytical techniques include: (1) the capability of using frequency response estimates of the plant generated either from experimental data or from an analytical model, (2) the ability to have some measure of control over the structure (e.g., decentralized) and order of the controller, (3) and the capability of simultaneously specifying several closed loop design constraints. The primary drawbacks to this approach are that the user must specify an initial, stabilizing controller and that there is no guarantee that the algorithm will converge to a solution that satisfies all the design constraints. However, under certain mild constraint feasibility assumptions this method can ensure that the "worst" constraint violation does improve at each iteration. The development of the Model and Data-Oriented Computer-Aided Design System (MADCADS), a menu-driven software program that uses this numerical method, is currently underway at Ohio University with support provided by NASA Marshall Space Flight Center.

The remainder of this section is outlined as follows. First, the properties of the SAGA-II controller are analyzed since it is the starting point for the numerical redesign. Second, the modifications made to the SAGA-II controller are presented. Finally, simulation results using the redesign controller are presented and compared with simulation results using SAGA-II.

6.1 Analysis of the SAGA-II Controller

The SAGA-II controller is the result of efforts by NASA to redesign the PCS in order to reduce undesirable pointing errors introduced by SA bending. Each of the diagonal terms of the controller consists of second-order PID compensation for rejection of low frequency torque disturbances, FIR filters for suppression of RWA isolator modes, and lightly damped poles near 0.1 Hz and 0.6 Hz for the rejection of torque disturbances generated by SA motion. All of the off-diagonal terms of the controller are zero. To illustrate the general characteristics of SAGA-II, the frequency response of the V_3 axis of the controller is shown in Fig. 6.1. The lightly damped poles and FIR filter are quite apparent. The frequency domain performance properties of the PCS using SAGA-II are illustrated by the magnitude frequency responses of the diagonal elements of the closed loop transfer function from the disturbance inputs to the measured LOS shown in Figs. 6.2-6.4. These responses reveal the deep but narrow notches at 0.1 Hz that are the result of the lightly damped controller poles that are intended to reduce the impact of the out-of-plane bending of the SA's on the LOS. The notches at 0.6 Hz which suppress the effects of in-plane SA bending are also apparent. The most important observation is the fact that in each of these plots there are several frequency regimes in which the level of disturbance rejection is not so pronounced as at 0.1 Hz and 0.6 Hz. As is to be shown, the overall level of disturbance suppression can be reduced by increasing the damping ratios of the controller poles that produce these notches. The responses of the diagonal elements have been chosen for illustration purposes because of the highly diagonally dominant nature of the system.

MIMO stability robustness of the system is indicated by the frequency dependent singular value plot of the complementary sensitivity function (a measure of unstructured multiplicative output uncertainty in the plant model) that has been generated using the TREETOPS model and is shown in Fig. 6.5. The peak values of this plot located between 1.0 Hz and 2.0 Hz indicate relatively poor stability robustness. The classical SISO stability margins given in Table 6.1 also indicate poor stability robustness.

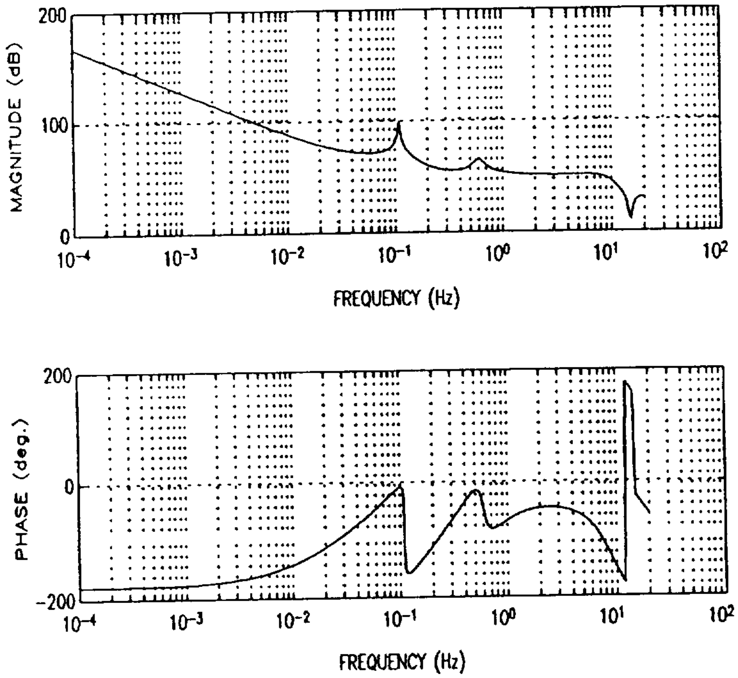

Figure 6.1 Bode Plot of V_3 Axis of SAGA-II Controller

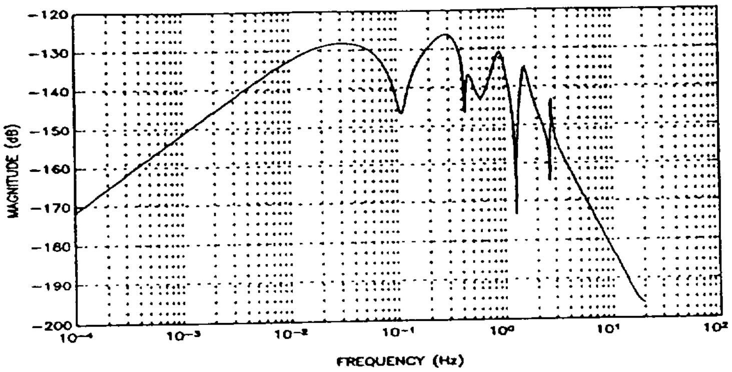

Figure 6.2 Magnitude of Frequency Response from Disturbance to LOS for the V_1 Axis with SAGA-II

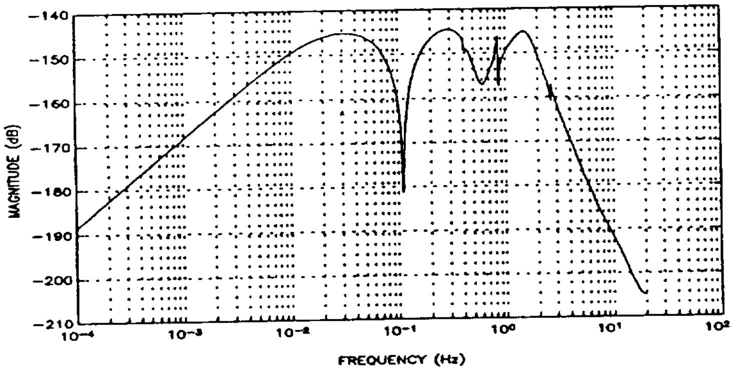

Figure 6.3 Magnitude of Frequency Response from Disturbance to LOS for V_2 Axis with SAGA-II

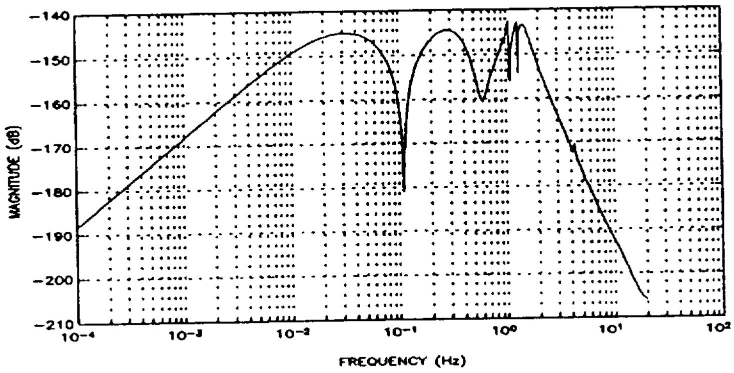

Figure 6.4 Magnitude of Frequency Response from Disturbance to LOS for V_3 Axis with SAGA-II

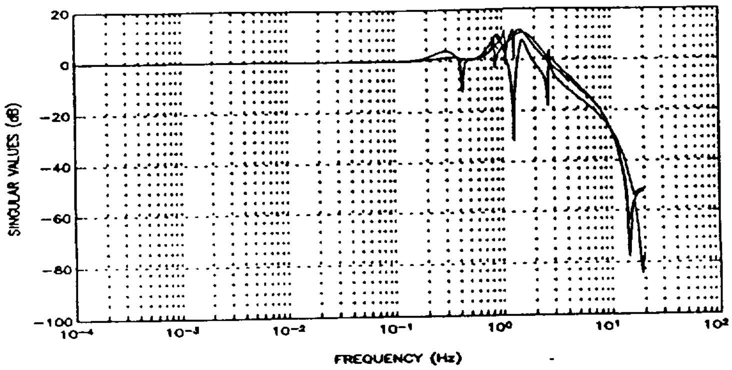

Figure 6.5 Singular Value Bode Plot of Complementary Sensitivity Function with SAGA II

Table 6.1 SISO Stability Margins with SAGA-II Controller

HST Axis	V_1	V_2	V_3
Phase Margin	25°	15°	15°
Lower Gain Margin	15dB	20dB	25dB
Upper Gain Margin	10dB	5dB	7dB

Linear simulation of the closed loop system using SAGA-II was performed using software and data provided by NASA. The model used for simulation was the 90° non-composite MIMO modal model described in Section 3. The results of the first 200 seconds from a 500 second linear simulation without reaction wheel torque limits using data for the 90° SA orientation are shown in Figs. 6.6-6.8. Simulation results with torque limits are shown in Figs. 6.12-6.14. PSD estimates corresponding to these two simulations are shown in Figs. 6.9-6.11 and Figs. 6.15-6.17, respectively. It apparent from these simulations that the results of simulation with torque limits are almost the same as that obtained from the linear simulation. The PSD estimate for the V_1 axis reveals that the disturbance power density at most frequencies below 2.0 Hz has a significant impact on this axis. The estimate for V_2 indicates significant disturbance power density in the vicinity of 0.1 Hz while the estimate for V_3 reveals that most of the power is between 0.2 Hz and 3.0 Hz with a peak near 0.6 Hz. The results of these simulations are in agreement with what is expected for a 90° SA orientation, i.e., the impact of the 0.1 Hz mode is greatest on the V_1 and V_2 axes, while the impact of the 0.6 Hz mode is greatest on the V_3 axis. It is important in all cases to recognize that while these power density spectra do indicate concentrations of power density near 0.1 Hz and 0.6 Hz, that there are also significant levels of power density at nearby frequencies.

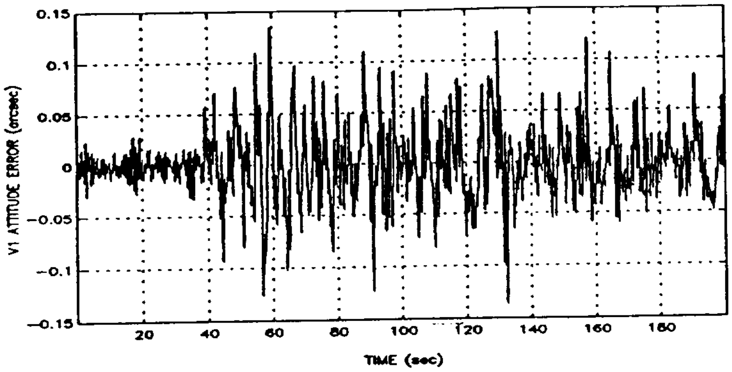

Figure 6.6 V_1 Linear Simulation Using SAGA-II

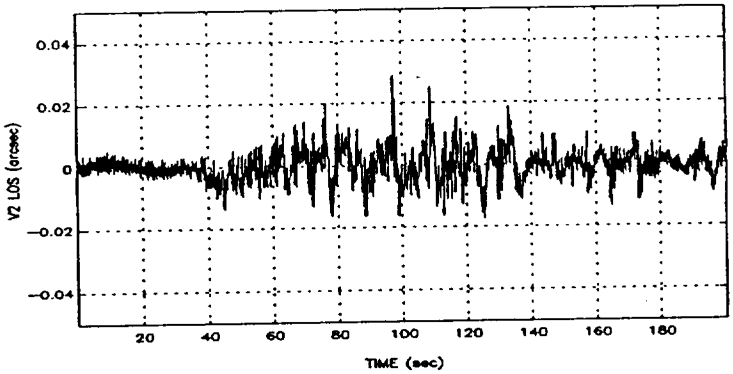

Figure 6.7 V_2 Linear Simulation Using SAGA-II

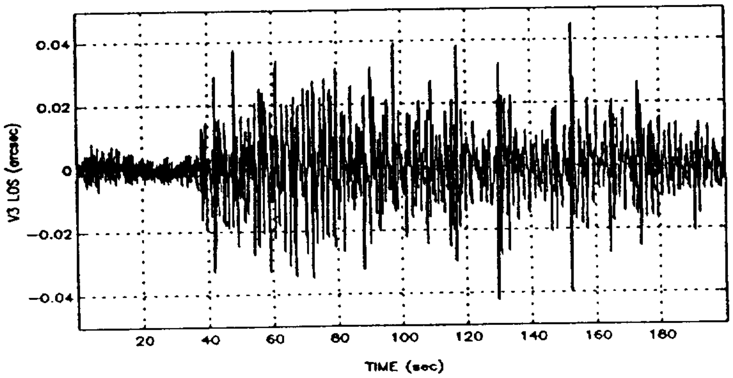

Figure 6.8 V_3 Linear Simulation Using SAGA-II

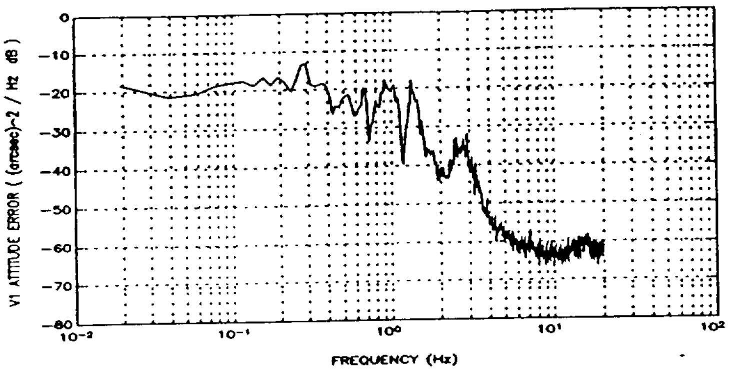

Figure 6.9 V$_1$ PSD Estimate from Linear Simulation Using SAGA-II

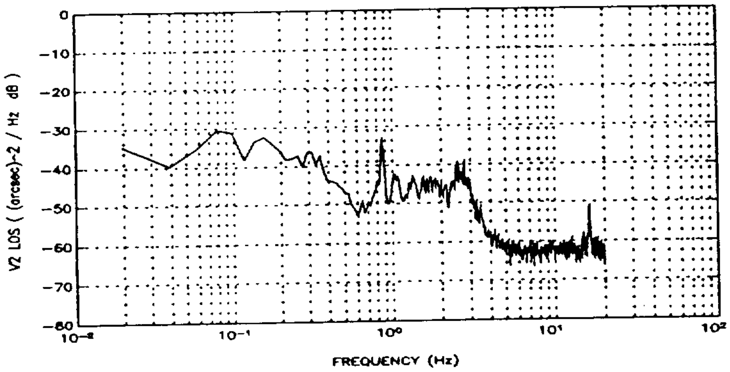

Figure 6.10 V$_2$ PSD Estimate from Linear Simulation Using SAGA-II

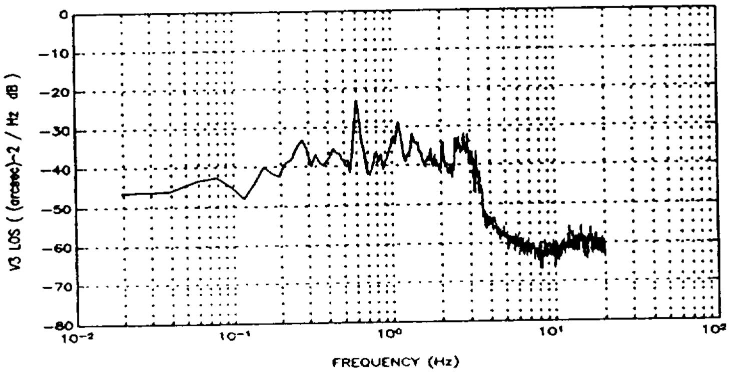

Figure 6.11 V$_3$ PSD Estimate from Linear Simulation Using SAGA-II

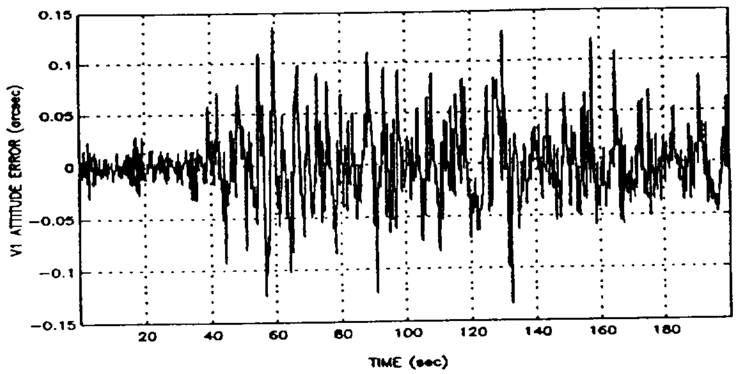

Figure 6.12 V_1 Simulation Using SAGA-II and Torque Limits

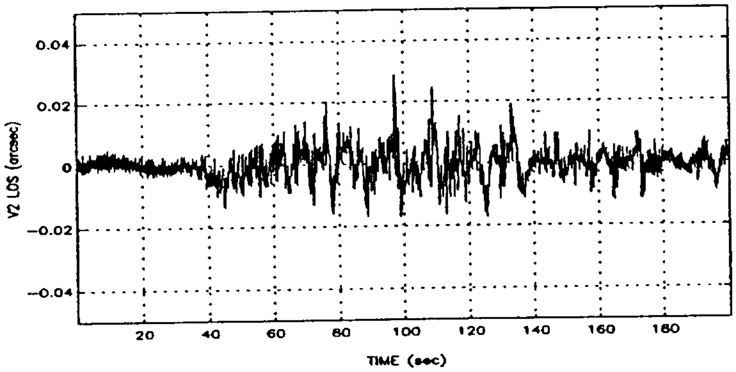

Figure 6.13 V_2 Simulation Using SAGA-II and Torque Limits

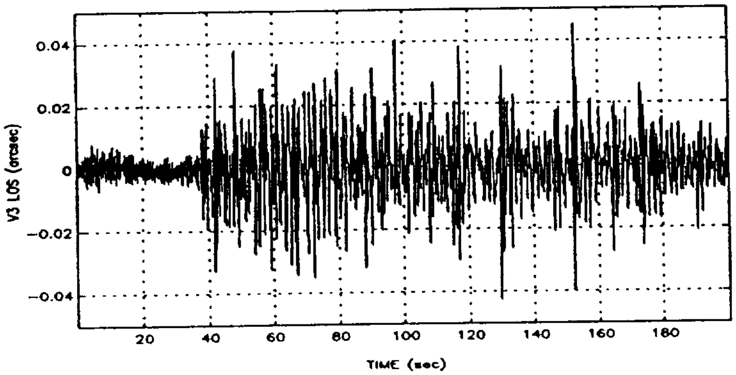

Figure 6.14 V_3 Simulation Using SAGA-II and Torque Limits

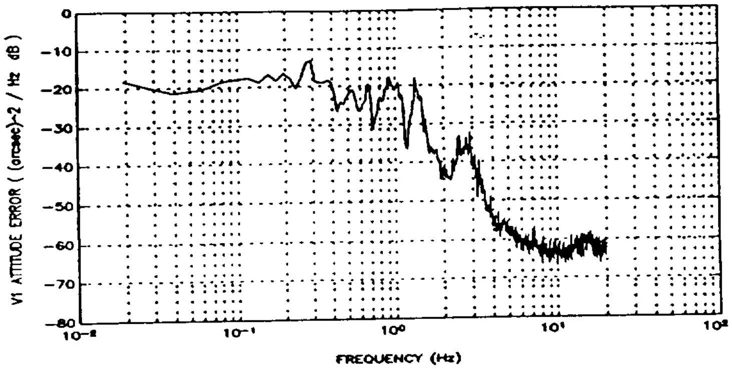

Figure 6.15 V_1 PSD Estimate from Simulation Using SAGA-II and Torque Limits

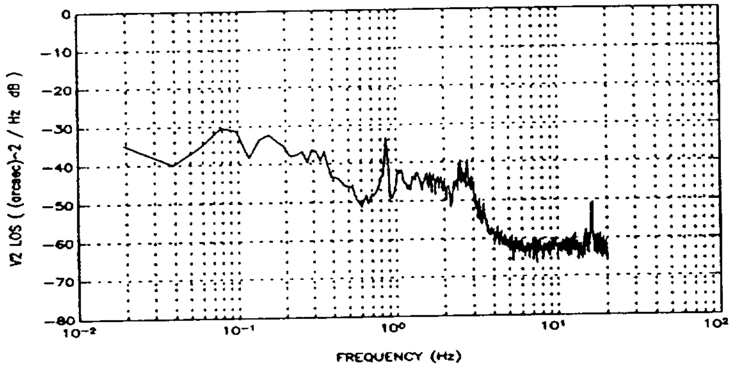

Figure 6.16 V_2 PSD Estimate from Simulation with SAGA-II and Torque Limits

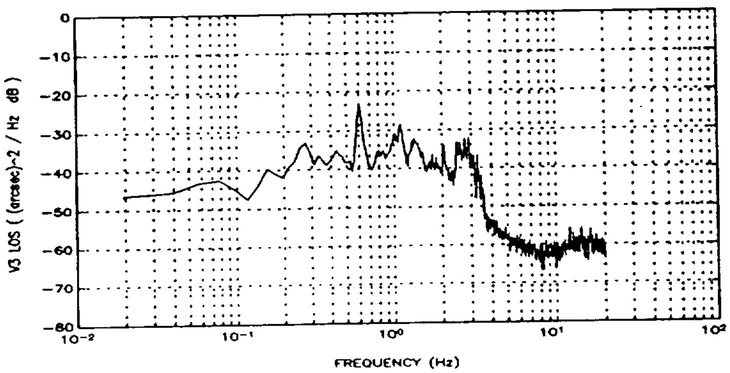

Figure 6.17 V_3 PSD Estimate from Simulation Using SAGA-II and Torque Limits

6.2 Procedure for Numerical Redesign

The block diagram shown in Fig. 6.18 was chosen as the basis for the numerical design process. The signals and blocks in the figure are defined as follows:

$\Delta\theta_{cmd}$: commanded change in attitude (commanded rate)
θ_{meas}: measured attitude
T_d: torque disturbances used to model impact of SA vibration
G_{fix}: fixed stage of the controller
G_{design}: designable stage of the controller
G_{delay}: computational delay of control computer
G_{Iv}: inertia matrix
G_{HST}: discretized model of the Hubble Space Telescope
G_{diff}: digital differentiators

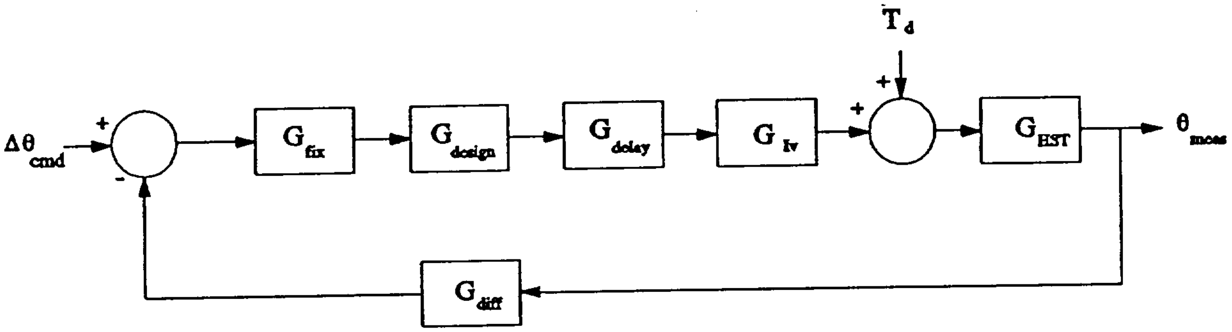

Figure 6.18 Block Diagram Used for Numerical Design

From this diagram the measured attitude as a function of the disturbance torque is calculated to be

$$\theta_{meas} = S_\theta G_{HST} T_d \tag{4.1}$$

where $S_\theta = \left(I + G_{HST} G_{Iv} G_{delay} G_{design} G_{fix} G_{diff}\right)^{-1}$ is the output sensitivity function.

Since it is desirable to reduce the impact of T_d on θ_{meas}, it is necessary to keep $S_\theta G_{HST}$ small in norm at frequencies where T_d contains significant power. This can be achieved in an indirect manner by designing the controller such that S_θ is small in norm at the appropriate frequencies. It is also desirable to keep the complementary sensitivity function $T_\theta = I - S_\theta$

small in norm at frequencies for which the model of the plant is suspected as being inaccurate (primarily high frequencies).

Due to the proven performance enhancements achieved by SAGA-II the decision was made to use the basic SAGA-II design as a starting point for the numerical redesign; however a few modifications were made before applying MADCADS. These modifications are now described below.

Analysis of the geometry of the HST indicates that regardless of SA orientation any in-plane bending of the SA's has only a minor impact on HST motion about the V_2 axis. Therefore, the lightly damped pole located at 0.6 Hz in the V_2 axis of the SAGA-II controller was eliminated. This allows for increased gain in this loop while maintaining closed loop stability. On the other hand, in-plane bending has a significant impact on the V_1 and V_3 axes; the degree depending on the particular SA orientation. The out-of-plane SA bending has an impact upon all HST axes, especially V_2; therefore, the frequencies of the other poles of SAGA-II were not changed. However, the damping ratios of these poles were increased in order to spread the disturbance rejection benefits that they offer over a broader range of frequencies. This change is based upon the fact that the PSD estimates from the simulation with SAGA-II reveal that the disturbance power density is not simply isolated to the frequencies near 0.1 Hz and 0.6 Hz, but is also present in significant levels at nearby frequencies. A final change was the inclusion of two stages of phase lead compensation in each of the primary axes of the controller in order to improve the stability robustness characteristics of the system. The high frequency FIR filters and PID compensation were left intact.

To set up the numerical redesign the frequency response of the HST was generated at 675 frequency points ranging from 10^{-4} Hz to 20 Hz using the TREETOPS model of the 90° SA orientation (this was the only MIMO model available at the time the numerical redesign). Frequency responses were also generated for the fixed stage of compensation (PID, disturbance suppression poles, and FIR filters). These responses were used as input to MADCADS along with the initial values for matrices corresponding to a state model of the sixth order phase lead stages, which were used as the free part of the controller. The frequency domain design constraints on the sensitivity function (for disturbance rejection) and complementary sensitivity function (for stability robustness) as shown in Figs. 6.19 and 6.20 were also used as input to MADCADS. The constraints on S_θ and T_θ were chosen in such a way as to attempt to maintain the disturbance attenuation of the initial controller with the objective of improving the stability robustness, especially at frequencies above 1.0 Hz. It is important to note that the current version of MADCADS allows for the inclusion of constraints that are far more sophisticated than those used here. For example, constraints on the shapes of the magnitude frequency responses of individual input-output pairs of various open-loop and closed-loop transfer functions can be included.

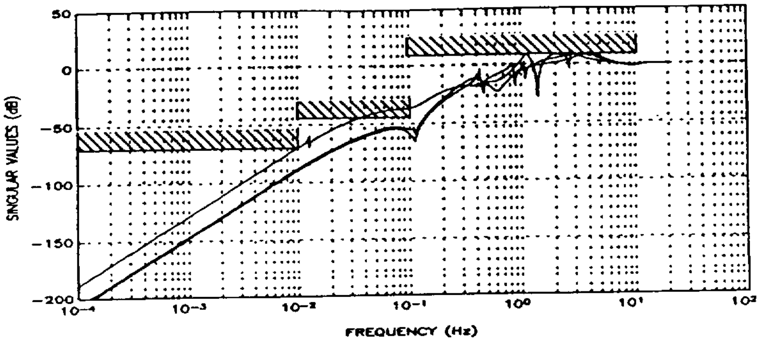

Figure 6.19 Singular Value Bode Plot of Initial Sensitivity Function and Constraints

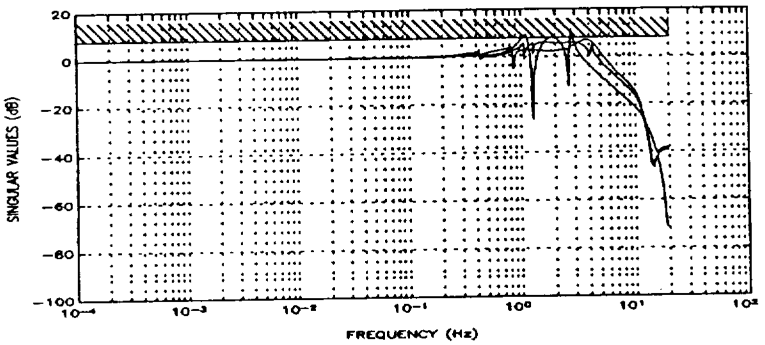

Figure 6.20 Singular Value Bode Plot of Initial Complementary Sensitivity Function and Constraints

The MADCADS program was run for 100 iterations in an attempt to achieve the design constraints. Final results are shown in Figs. 6.21 and 6.22. As these figures illustrate, no dramatic improvements over the initial controller were achieved, although some improvement in stability robustness was realized. It is suspected that this inability to make large improvements is primarily due to the fact that the free part of the controller design is only sixth order and that the constraints were in constant competition, i.e., the only way to improve the stability robustness is to weaken the disturbance rejection constraint. Classical SISO stability robustness measures are given in Table 6.2. When compared to SAGA-II (Fig 6.5 and Table 6.1) however, these results reveal significant improvements in the stability robustness characteristics of the system.

The magnitude frequency responses of the diagonal elements of the transfer function matrix from the disturbance inputs to the LOS for the closed loop system with the final controller are shown in Figs. 6.23-6.25. Comparison of these plots with those in Figs. 6.2-6.4 indicates considerable improvements in disturbance rejection over those achieved by SAGA-II.

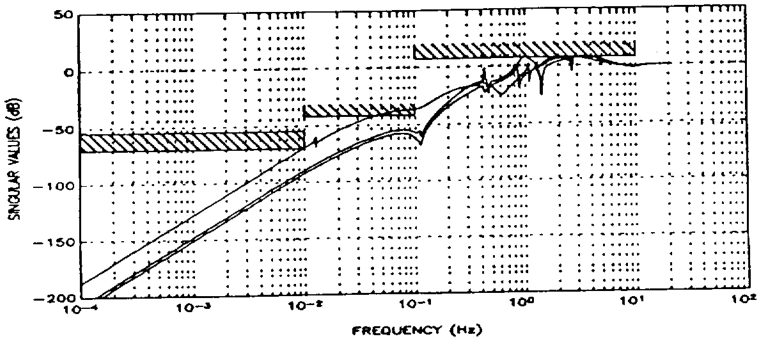

Figure 6.21 Singular Value Bode Plot of Final Sensitivity Function and Constraints

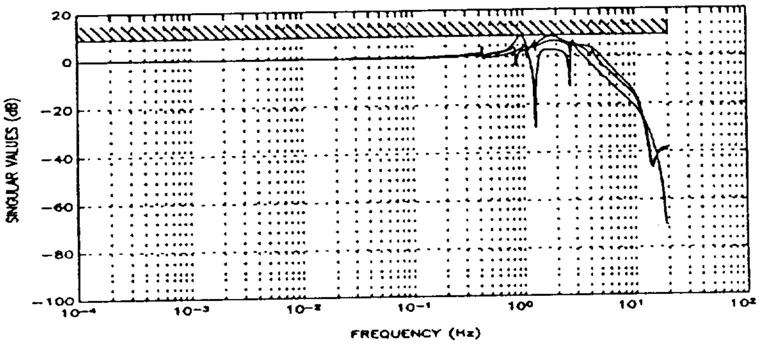

Figure 6.22 Singular Value Bode Plot of Final Complementary Sensitivity Function and Constraints

Table 6.2 SISO Stability Margins Corresponding to Final Controller

HST Axis	V_1	V_2	V_3
Phase Margin	30°	40°	30°
Lower Gain Margin	50dB	15dB	20dB
Upper Gain Margin	7dB	7dB	7dB

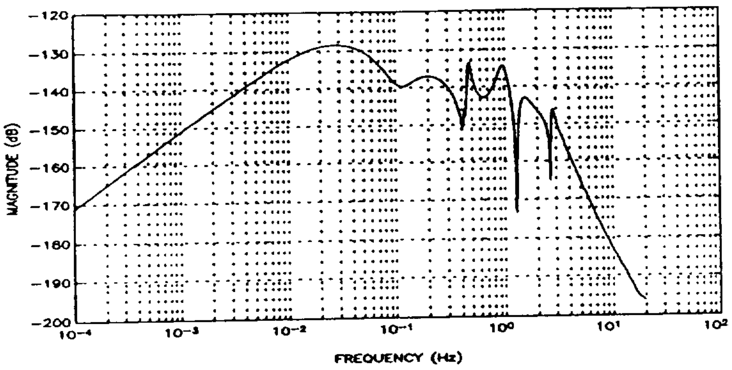

Figure 6.23 Magnitude of Frequency Response from Disturbance to LOS for V_1 with Redesigned Controller

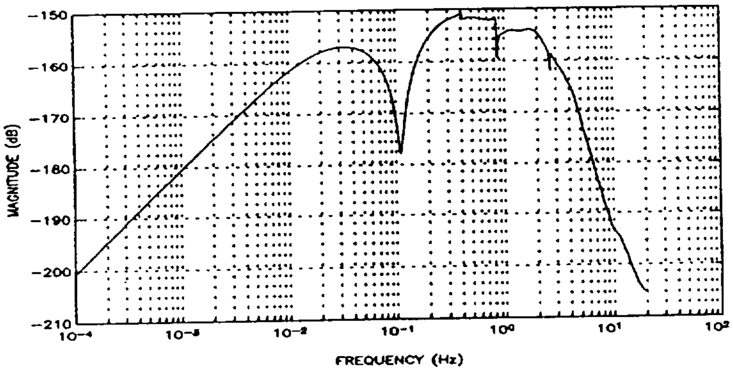

Figure 6.24 Magnitude of Frequency Response from Disturbance to LOS for V_2 with Redesigned Controller

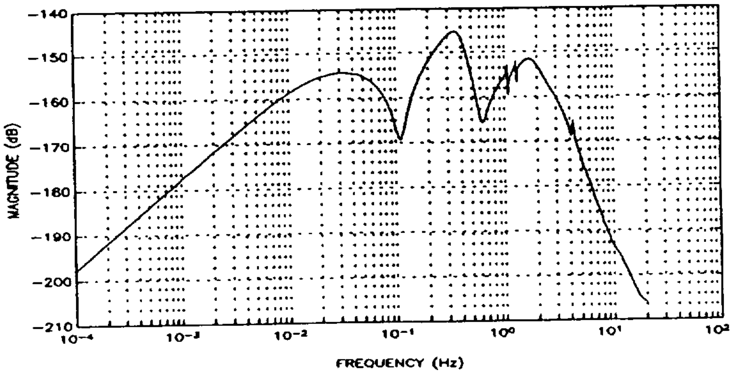

Figure 6.25 Magnitude of Frequency Response from Disturbance to LOS for V_3 with Redesigned Controller

6.3 Simulation Results

Simulations with and without reaction wheel torque limits (0.8 N-m) were performed with the redesigned controller in the loop. Results are given in Figs. 6.26-6.37 and Tables 6.3-6.6. All the plots contain data generated using the modified FORTRAN simulation discussed in Section 4 with the 90° non-composite MIMO modal model. The PSD estimates are generated from 500 second simulations, even though the time responses are given only out to 200 seconds. The tables contain data derived from simulations using both the 90° non-composite MIMO modal model and the TREETOPS model. In a large majority of cases the redesigned controller proved to have better peak and RMS performance than SAGA-II, especially for the simulations using the TREETOPS model. The general performance improvements can be attributed to the increased disturbance rejection at nearly all frequencies of interest as illustrated by Figs. 6.23-6.25. As stated previously, this increased rejection is obtained by eliminating an unnecessary component of SAGA-II, namely the lightly damped pole in the V_2 axis and increasing the damping ratios of the other controller poles in order to spread out the disturbance rejection near 0.1 Hz and 0.6 Hz. These simulations provide an excellent example of how "overtuning" a controller to a "known" disturbance can lead to inferior performance. This is especially true in the case of plant and disturbance models possessing lightly damped dynamics.

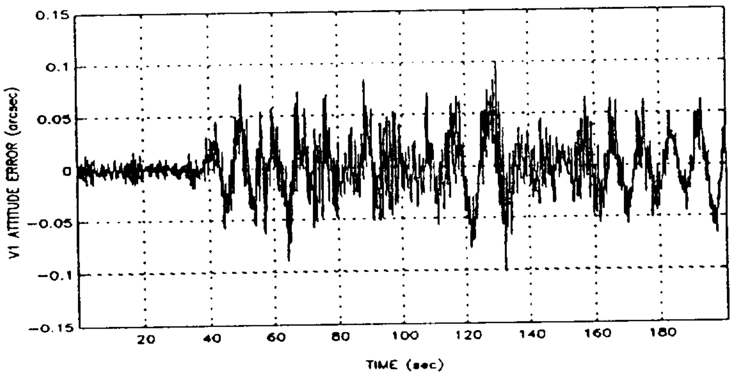

Figure 6.26 V_1 Linear Simulation Using Redesigned Controller

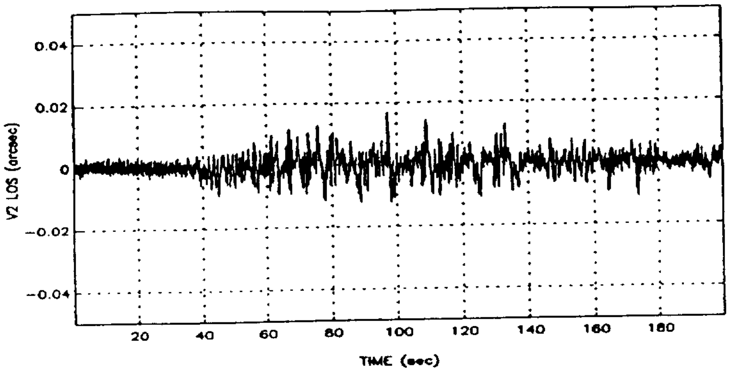

Figure 6.27 V_2 Linear Simulation Using Redesigned Controller

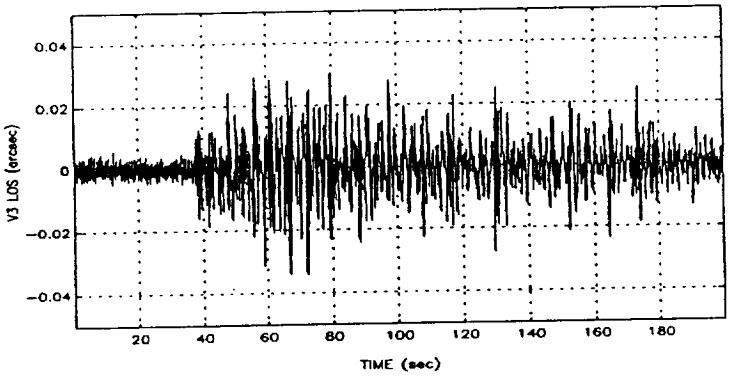

Figure 6.28 V_3 Linear Simulation Using Redesigned Controller

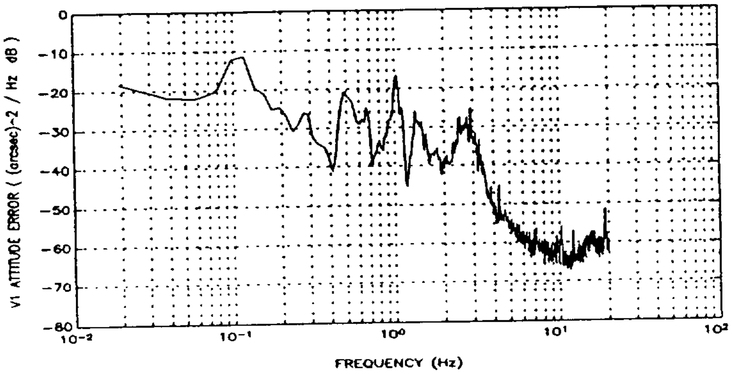

Figure 6.29 V_1 PSD Estimate from Linear Simulation Using Redesigned Controller

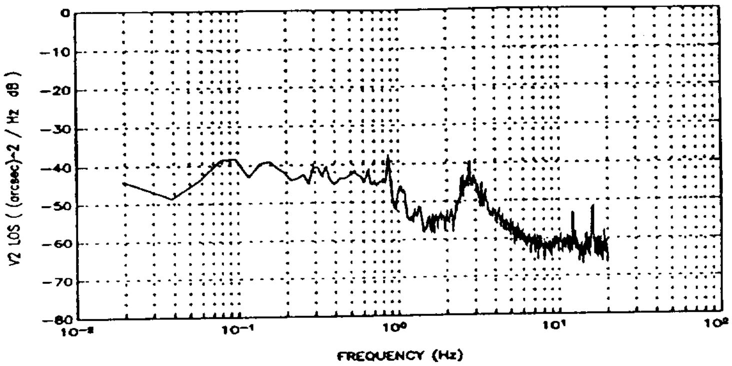

Figure 6.30 V_2 PSD Estimate from Linear Simulation Using Redesigned Controller

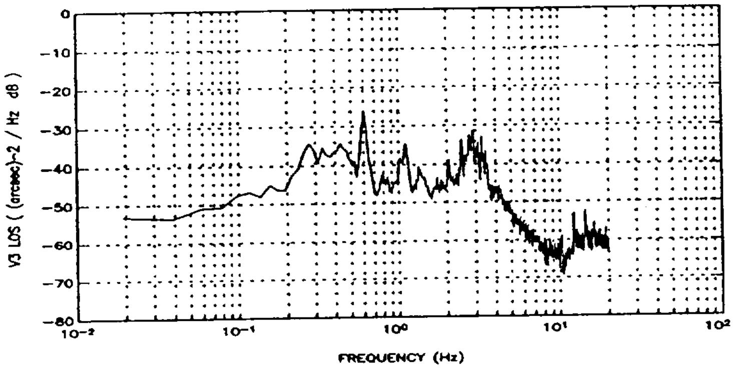

Figure 6.31 V_3 PSD Estimate from Linear Simulation Using Redesigned Controller

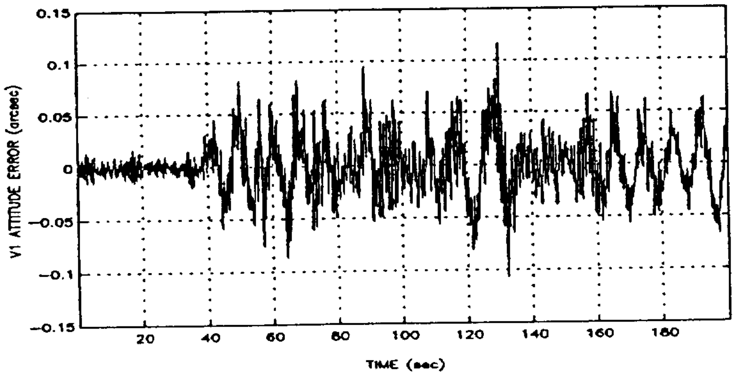

Figure 6.32 V_1 Simulation Using Redesigned Controller and Torque Limits

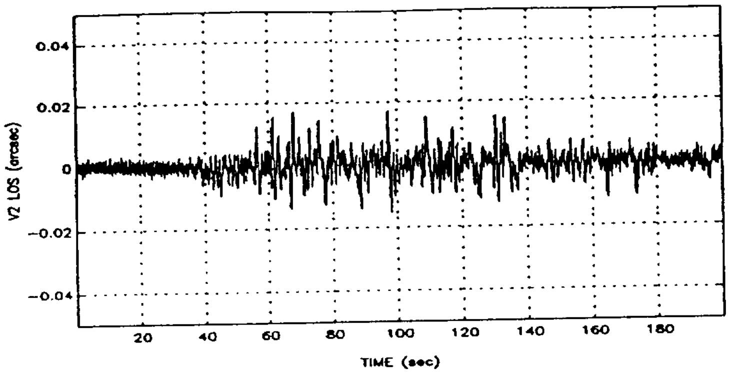

Figure 6.33 V_2 Simulation Using Redesigned Controller and Torque Limits

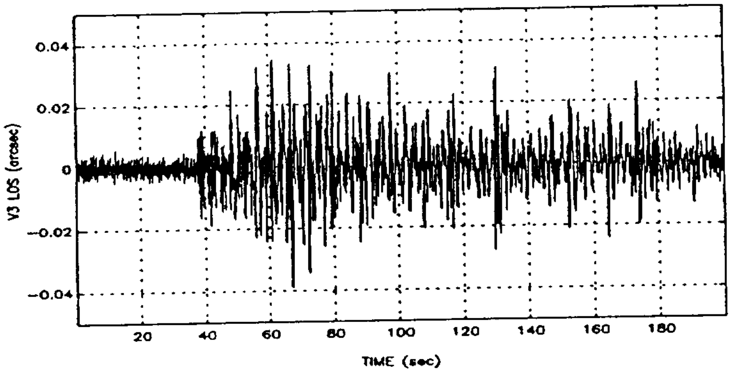

Figure 6.34 V_3 Simulation Using Redesigned Controller and Torque Limits

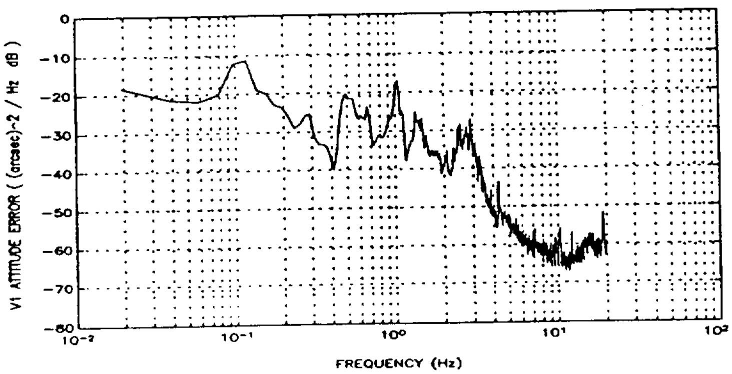

Figure 6.35 V_1 PSD Estimate from Simulation Using Redesigned Controller and Torque Limits

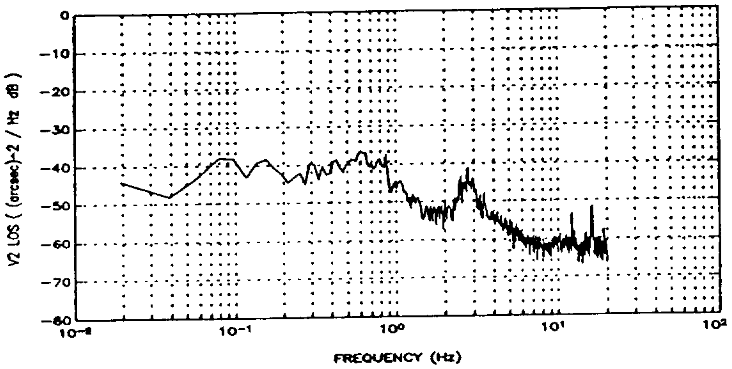

Figure 6.36 V_2 PSD Estimate from Simulation Using Redesigned Controller and Torque Limits

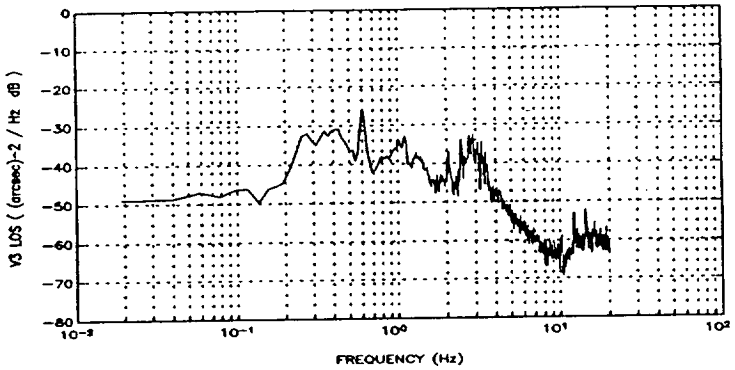

Figure 6.37 V_3 PSD Estimate from Simulation Using Redesigned Controller and Torque Limits

Table 6.3 Peak and RMS Attitude Values for Linear Simulation with Non-Composite Modal Model

HST Axis		V_1	V_2	V_3
Peak Value (milliarcsec)	SAGA II	136.8	28.9	67.8
	Redesign	114.8	18.9	57.8
RMS Value (milliarcsec)	SAGA II	26.4	3.44	6.67
	Redesign	21.1	2.33	5.03

Table 6.4 Peak and RMS Attitude Values for Linear Simulation with TREETOPS Model

HST Axis		V_1	V_2	V_3
Peak Value (milliarcsec)	SAGA II	170.9	28.9	68.0
	Redesign	100.9	17.4	61.0
RMS Value (milliarcsec)	SAGA II	32.4	3.36	7.11
	Redesign	23.1	2.25	5.27

Table 6.5 Peak and RMS Attitude Values for Simulation with Non-Composite Modal Model and Torque Limits

HST Axis		V_1	V_2	V_3
Peak Value (milliarcsec)	SAGA II	136.3	28.8	75.7
	Redesign	153.7	25.2	76.2
RMS Value (milliarcsec)	SAGA II	26.6	3.44	6.75
	Redesign	21.5	2.60	5.54

Table 6.6 Peak and RMS Attitude Values for Simulation with TREETOPS Model and Torque Limits

HST Axis		V_1	V_2	V_3
Peak Value (milliarcsec)	SAGA II	171.5	28.9	74.2
	Redesign	123.9	20.9	70.4
RMS Value (milliarcsec)	SAGA II	32.5	3.40	7.34
	Redesign	23.9	2.48	5.67

Simulations with reaction wheel torque limits and the fixed point arithmetic implementation described in Section 4.4 were performed using the redesigned controller. Plots of the time response and PSD estimates are presented in Figs. 6.38-6.43. These simulations were performed using the 90° non-composite MIMO modal model. Tabular data is provided for simulations using both the 90° non-composite MIMO modal model and the TREETOPS model in Tables 6.7 and 6.8, respectively. As is clearly illustrated by these plots and data, performance degradations do occur with the simulation of fixed point arithmetic. It is important to realize that: (1) these simulations are for a rather arbitrary state-space realization for the controller, (2) that different realizations can produce widely varying simulation results, and (3) this does not necessarily reflect how this controller would perform as an HST on-board implementation. Therefore, before any controller is actually implemented attempts must be made to find a controller realization that reduces the effects of fixed point arithmetic.

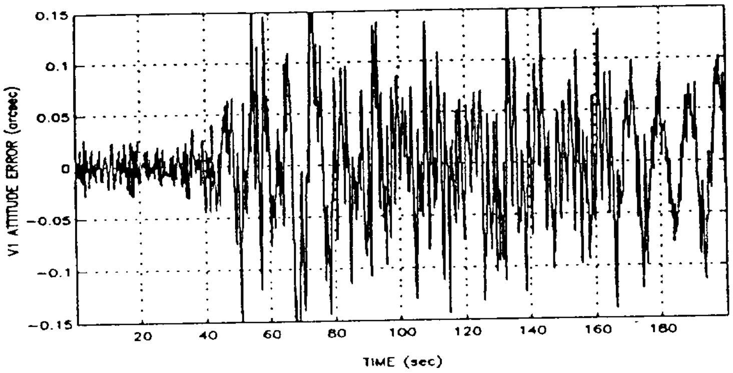

Figure 6.38 V_1 Simulation Response Using Redesigned Controller, Torque Limits, and Fixed Point Arithmetic

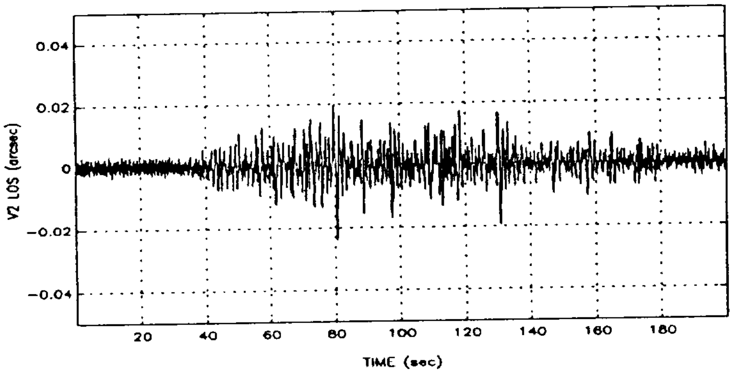

Figure 6.39 V_2 Simulation Response Using Redesigned Controller, Torque Limits, and Fixed Point Arithmetic

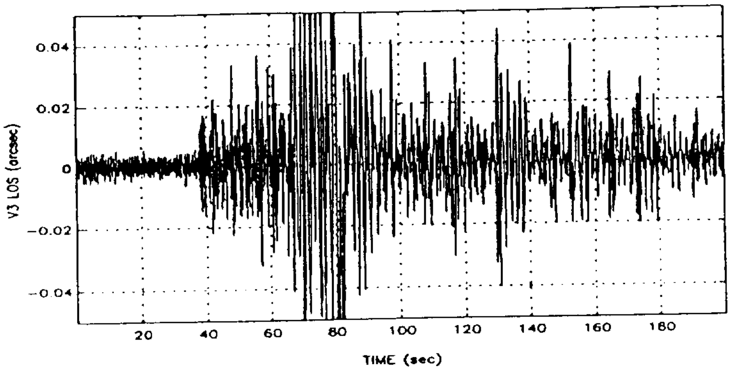

Figure 6.40 V_3 Simulation Response Using Redesigned Controller, Torque Limits, and Fixed Point Arithmetic

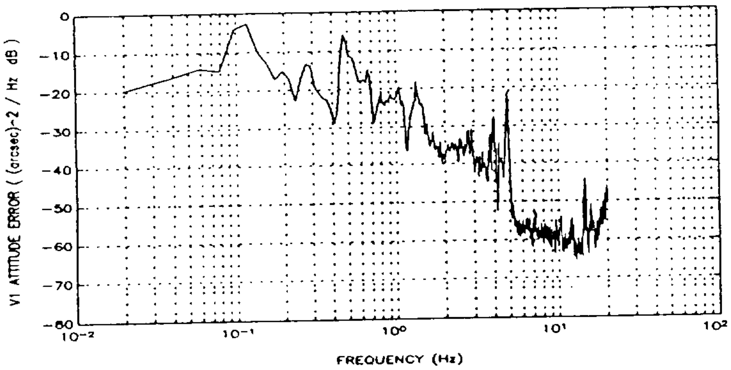

Figure 6.41 V_1 PSD Estimate from Simulation Using Redesigned Controller, Torque Limits, and Fixed Point Arithmetic

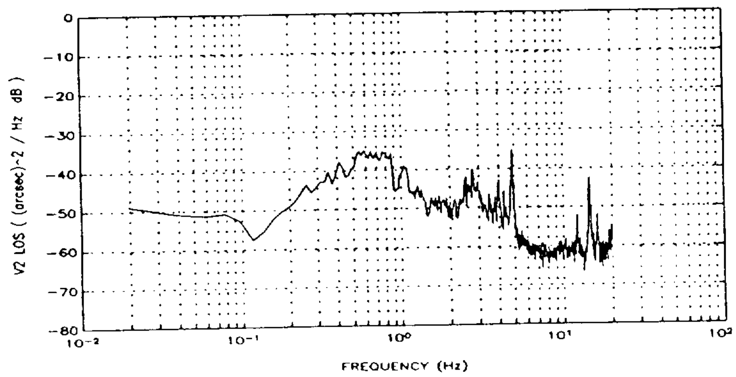

Figure 6.42 V_2 PSD Estimate from Simulation Using Redesigned Controller, Torque Limits, and Fixed Point Arithmetic

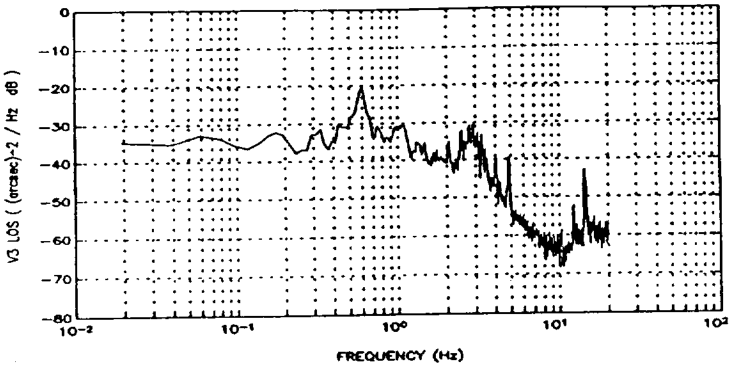

Figure 6.43 V_3 PSD Estimate from Simulation Using Redesigned Controller, Torque Limits, and Fixed Point Arithmetic

Table 6.7 Peak and RMS Attitude Values for Simulation with Non-Composite Modal Model, Torque Limits and Fixed Point Arithmetic

HST Axis		V_1	V_2	V_3
Peak Value (milliarcsec)	Redesign	198.0	27.2	110.0
RMS Value (milliarcsec)	Redesign	47.8	3.62	10.8

Table 6.8 Peak and RMS Attitude Values for Simulation with TREETOPS Model, Torque Limits and Fixed Point Arithmetic

HST Axis		V_1	V_2	V_3
Peak Value (milliarcsec)	Redesign	440.8	43.8	132.9
RMS Value (milliarcsec)	Redesign	45.1	3.36	8.80

7.0 CONCLUSIONS AND RECOMMENDATIONS

Efforts to redesign the controller for the Hubble Space Telescope Pointing Control System to achieve improved performance in the face of Solar Array induced disturbances have been presented. Controller designs via standard analytical H^∞ techniques led to a successful 82nd order controller that provides marked improvement in linear simulation results over the results obtained using SAGA-II; however, this controller results in an unstable response when reaction wheel torque limits are included or when a plant model other than the design model is used for simulation. Reduced-order controllers obtained from the 82nd order H^∞ controller via standard model reduction techniques result in an unstable closed loop.

A successful 28th order controller that provides simulation performance superior to that of SAGA-II was designed via an iterative numerical technique. This controller also performs well when a plant model other than the design model is used and when reaction wheel torque limits are included. A key reason for the success of this controller is the broad band disturbance rejection characteristics of the closed loop system. SAGA-II, on the other hand, places a greater emphasis on narrow band attenuation at 0.1 Hz and 0.6 Hz at expense of attenuation of disturbances at other frequencies.

Another area in which significant work was performed was modification of the FORTRAN simulation software provided by NASA. The simulation has been modified to run on a Sun SPARC® 2 workstation under FORTRAN 77 in order to reduce execution time. The G_{c1} and G_{c2} controllers of the simulation have been changed from the hard-coded PID and SAGA-II controllers to general state-space controller realizations, allowing the use of any controller in the simulation. Torque limits and a general fixed point arithmetic implementation have been included.

Several recommendations are made in regard to future Guest Investigator Programs. A meeting between NASA personnel and Guest Investigators at the initiation of a project to establish and clarify design goals would help the Investigators to more efficiently utilize their time. Provision of clear and complete information and verified, working software in the project information package would provide an ideal starting point for such an effort. In fact, in the HST PCS controller redesign project some of the greatest efforts expended related to understanding the design goals and modifying the software provided by NASA.

Some recommendations regarding MIMO controller design efforts are: (1) either provide a minimal MIMO design model or the data necessary to construct one, (2) provide simulation software able to handle arbitrary controllers and general fixed point arithmetic, and (3) be aware of the fact that modern, MIMO controller design techniques typically cannot produce low-order controllers when used with high order plant models. An alternative approach is to produce an initial low order stabilizing controller using classical methods followed by controller tuning with an iterative numerical technique. This approach can yield a low order controller regardless of the plant model order.

REFERENCES

1. Sharkey, J.P., Nurre, G.S., Beals, G.A., and Nelson, J.D., "A Chronology of the On-Orbit Pointing Control System Changes on the Hubble Space Telescope and Associated Improvements," *Proceedings of the 1992 AIAA Conference on Guidance, Navigation, and Control*, Hilton Head, SC, August, 1992.

2. Sills, J.W., Jr., "HST On-Orbit Structural Dynamics Models and Modal Gain Factors of October 1991", Lockheed Engineering Memorandum SPS 657, Nov. 9, 1991.

3. Follett, R.F., "Emulation and Characterization of Digital Controllers Realized with Fixed Point Arithmetic", M.S. Thesis, Mississippi State University, Mississippi State, MS, May, 1984.

4. Irwin, R.D., Frazier, W.G., and Medina, E., NASA Marshall Space Flight Center, "Controller Design for Large Flexible Aerospace Structures via Numerical Techniques for Achieving Multi-objective Design Specifications", Final Report NAG8-123 (Task 21), June, 1992.

5. King, J.A. and Irwin, R.D., "Issues in the Application of H^∞ to Large Space Structures", 22nd Southeastern Symposium on System Theory, Cookeville, TN, March 1990.

6. Maciejowski, J.M., *Multivariable Feedback Design,* Addison-Wesley, 1989.

7. Glover, K., "All Optimal Hankel-Norm Approximations of Linear Multivariable Systems and their L-Infinity Error Bounds,", *International Journal of Control,* Vol 39, No.6, pp. 1115-1193.

8. Chiang, R.Y. and Safanov, M.G., *MATLAB Robust Control Toolbox Manual,* The Mathworks, Inc., 1988.

9. W. Garth Frazier and R. Dennis Irwin, "Designing Reduced-Order Linear Multivariable Controllers Using Experimentally Derived Plant Data," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 16, No. 1, Jan-Feb, 1993.